AssetCenter™

Version 3.5

Référence de programmation

28 mars 2000

ITEM ACT-3.5X-FR-00701

Peregrine SYSTEMS®

The Infrastructure Management Company™

"Programmer's Reference"

Intégrité des données AssetCenter et InfraCenter for Workgroups

AssetCenter et InfraCenter for Workgroups sont des logiciels d'une grande richesse fonctionnelle. Cette richesse requiert l'utilisation d'une base de données de structure complexe : la base contient un nombre important de tables, champs, liens et index ; certaines tables intermédiaires ne sont pas affichées par l'interface graphique ; certains liens, champs et index sont automatiquement créés, supprimés ou modifiés par le logiciel.

Seules les interfaces prévues par AssetCenter et InfraCenter for Workgroups (interface graphique, API, programme d'importation, interface WEB, passerelles) sont à même de modifier le contenu de la base de données tout en respectant son intégrité. **Vous ne devez jamais modifier la structure ou le contenu de la base de données par d'autres moyens que ceux prévus par le logiciel** ; de telles modifications ont une forte probabilité d'entraîner la corruption de la base de données avec des manifestations telles que : perte ou modification involontaire de données ou de liens, création de liens ou enregistrements fantômes, messages d'erreur graves, etc.). Les altérations de la base de données résultant de ce type de manipulations entraînent la résiliation de la garantie et du support technique fournis par Peregrine Systems.

Environnements supportés par AssetCenter et InfraCenter for Workgroups

AssetCenter et InfraCenter for Workgroups fonctionnent dans un ensemble défini d'environnements dont la liste figure dans le manuel intitulé "Manuel d'installation et de mise à jour". L'utilisation d'AssetCenter et d'InfraCenter for Workgroups dans d'autres environnements que ceux prévus s'effectue aux risques de l'utilisateur. Les altérations de la base de données résultant de l'utilisation d'AssetCenter ou InfraCenter for Workgroups dans des environnements non prévus entraînent la résiliation de la garantie et du support technique fournis par le groupe Peregrine Systems.

Avant propos

Ce manuel a pour objectifs de :

- fournir à l'utilisateur une référence exhaustive des fonctions utilisables dans l'environnement AssetCenter.
- décrire la structure de la bibliothèque de fonctions, également appelée AssetCenter APIs. Cette bibliothèque permet d'écrire rapidement des applications clientes d'une base de données AssetCenter au moyen d'outils de développement tels que Visual Basic™ ou Access™.

Ce manuel est destiné en priorité aux lecteurs possédant une solide expérience en programmation.

Contacter Peregrine Systems

Siège mondial

Peregrine Systems, Inc.
3611 Valley Centre Drive
San Diego, CA 92130
Etats-Unis
Tél. : +1 858 481 5000 ou 800 638 5231
Fax : +1 858 481 1751
Web: http://www.peregrine.com

Support Clients :
Tél. : +1 858 794 7402 ou 800 960 9998
Fax : +1 858 794 6028
EMail : support@peregrine.com
Web : http://support.peregrine.com
Ouverture : du lundi au vendredi, de 05:00 à 17:30 (heures PST)

France, Espagne, Grèce et Afrique (sauf Afrique du Sud)

Peregrine Systems
Tour Franklin - La Défense 8
92042 Paris - La Défense Cedex
France
Tél. : +33 (0)1 47 73 11 11
Fax : +33 (0)1 47 73 11 12

Support Clients :
Tél. : +33 (0) 800 505 100
Fax : +33 (0)1 47 73 11 61
EMail : frsupport@peregrine.fr
Ouverture : du lundi au vendredi, de 08:00 à 18:00 (heures locales)

Allemagne et Europe de l'Est

Peregrine Systems GmbH
Bürohaus ATRICOM
Lyoner Strasse 15

60528 Frankfurt
Allemagne
Tél. : +49 (0)(69) 6 77 34-0
Fax : +49 (0)(69) 66 80 26-26

Support Clients :
Tél. : 0800 2773823
Fax : +49 (0) (69) 66 80 26-26
EMail : psc@peregrine.de
Ouverture : du lundi au vendredi, de 08:00 à 17:00 (heures locales)

Royaume-Uni

Peregrine Systems, Ltd.
Ambassador House
Paradise Road
Richmond
Surrey TW9 1SQ
Royaume-Uni
Tél : +44 (0)181 332 9666
Fax : +44 (0)181 332 9533

Support Clients :
Tél : +44 (0)181 334 5844 ou 0800 834 7700
Fax : +44 (0)181 334 5890
EMail : uksupport@peregrine.com
Ouverture : du lundi au vendredi, de 08:00 à 18:00 (heures locales)

Danemark, Norvège, Finlande et Islande

Peregrine Systems A/S
Naverland 2, SAL
DK-2600 Glostrup
Danemark
Tél : +45 43 46 76 76
Fax : +45 43 46 76 77

Support Clients :
Tél. : +45 77 31 77 76
Fax : +45 43 46 76 77
EMail : support.nordic@peregrine.com
Ouverture : du lundi au vendredi, de 08:30 à 04:30 (heures locales)

Pays Bas, Belgique et Luxembourg

Peregrine Systems BV
Botnische Golf 9a
Postbus 244
3440 AE Woerden
Pays Bas
Tél : +31 (0) 348 43 7070
Fax : +31 (0) 348 43 7080

Support Clients :
Tél. : 0800 0230889 (Pays Bas)
ou 0800 74747575 (Belgique et Luxembourg)
Fax : +31 (0) 348 43 7080
EMail : benelux.support@peregrine.com
Ouverture : du lundi au vendredi, de 08:00 à 18:00 (heures locales)

Singapour

Peregrine Systems Pte.Ltd
#03-16
CINTECH III
77 Science Park Drive
Singapore Science Park
118256
Singapour
Tél : +65 778 5505
Fax : +65 777 3033

Italie

Peregrine Systems, S.r.l.
Via Cernaia, 2
20121 Milano
Italie
Tél : +39 (02) 6556931

Japon

Peregrine Systems K.K.
Level 32, Shinjuku Nomura Building
1-26-2 Nishi-shinjuku, Shinjuku-ku

Tokyo 163-0532
Japon

Tél : +81 (3) 5322-1350
Fax : +81 (3) 5322-1352

Support Clients :
Tél. : +81 (3) 5322-1350
Fax : +81 (3) 5322-1352
EMail : glipper@Peregrine.com

Suède

Peregrine Systems AB
Frösundaviks Allé 15, 4th floor
S-169 70 Solna
Suède
Tél : +46 (0)8-655 36 04
Fax : +46 (0)8-655 26 10

Support Clients :
Tél. : +45 77 31 77 76
Fax : +45 43 46 76 77
EMail : nordic@peregrine.com
Ouverture : du lundi au vendredi, de 08:30 à 04:30 (heures locales)

Faites nous part de vos commentaires

Notre objectif est de mettre à votre disposition les documentations les plus à jour et les plus utiles pour vous.

Vos commentaires sont les bienvenus.

N'hésitez pas à nous faire part de vos remarques en les adressant à **documentation@peregrine.com**.

Conventions utilisées

Les mises en forme suivantes ont des significations particulières :

Police fixe	Commande DOS.
`Exemple`	Exemple de code ou de commande.
...	Bout de code ou de commande omis.
Nom d'objet	Les noms de champs, d'onglets, de menus, de fichiers sont imprimés en gras.
Note	Note importante.

Les commandes sont décrites avec les conventions de notation suivantes :

[ ]	Ces crochets encadrent un paramètre optionnel. Ne les tapez pas dans votre commande. Exception : dans les scripts BASIC, les crochets qui encadrent le chemin d'accès à des données de la base doivent figurer dans le script : **[Lien.Lien.Champ]**
< >	Ces crochets encadrent un paramètre décrit en langage clair. Ne les tapez pas dans votre commande et remplacez le texte qu'ils encadrent par l'information qui doit y figurer.
{ }	Ces accolades encadrent des paramètres parmi lesquels un seul doit être choisi. Ne tapez pas les accolades dans votre commande
|	La barre verticale sépare les paramètres possibles qui figurent entre les accolades.
*****	L'astérisque ajoutée à droite de crochets indique que la formule qu'ils encadrent peut être répétée plusieurs fois.

Table des matières

Chapitre 1 - Introduction

Vous trouverez dans cette partie les informations suivantes :

- *Classification des fonctions*
- *Conventions*
- *Définitions*
- *Typage des fonctions et des paramètres de fonctions*

Classification des fonctions

La classification des fonctions s'effectue à trois niveaux différents. Une fonction donnée peut être classée par :

- *Familles de fonctions*
- *Champs d'application des fonctions*
- *Domaines fonctionnels*

Familles de fonctions

Les fonctions disponibles dans l'environnement AssetCenter peuvent être regroupées en trois grandes familles :

- les fonctions reconnues par AssetCenter : il s'agit essentiellement des fonctions utilisables dans les parties scriptables (en Basic) du logiciel.
- les fonctions reconnues par la bibliothèque AssetCenter APIs : ces fonctions peuvent être appelées par des outils externes ou par un programme écrit dans un langage évolué.
- les fonctions reconnues par la bibliothèque API d'AssetCenter Web : ces fonctions sont utilisables dans les sections script des modèles HTML.

Ces trois grandes familles de fonctions se recoupent. Par exemple, certaines fonctions des APIs AssetCenter sont utilisables dans les scripts Basic du logiciel. On dit alors qu'une telle fonction, provenant à l'origine des APIs AssetCenter, est "exposée" dans les scripts Basic internes à AssetCenter. Si la syntaxe d'une telle fonction peut alors varier sensiblement, son comportement reste inchangé.

Champs d'application des fonctions

Les fonctions décrites dans ce document sont utilisables dans au moins un des contextes suivants :

- Toutes les fenêtres de script d'AssetCenter. On parle alors de fonction "built-in".
- AssetCenter APIs
- Script de configuration d'un champ ou d'un lien (menu contextuel Configurer l'objet ou AssetCenter Database Administrator) et par extension Script de calcul (Nom SQL : memScript) d'un champ calculé :
  - √ Valeur par défaut.
  - √ Obligation de saisie.
  - √ Historisation.
  - √ Lecture seule.
- Script de configuration d'un paramètre de caractéristique :
  - √ Valeur par défaut.
  - √ Disponibilité.
  - √ Forcer l'affichage.
  - √ Obligation de saisie.
  - √ Historisation.
- Actions de type Script :
  - √ Script défini dans le champ Script de l'action (Nom SQL : Script) d'une action Script.
- Assistants AssetCenter :
  - √ Script "FINISH.DO" d'un assistant.
  - √ Script de définition des valeurs des propriétés d'un nœud.
- Librairies API AssetCenter.

Domaines fonctionnels

Chaque fonction est associée à un domaine fonctionnel. Ce domaine fonctionnel décrit la nature des opérations effectuées par la fonction. Les domaines fonctionnels sont les suivants :

- Connexion
- Gestion des objets tables, requêtes et enregistrements
- Gestion des objets requêtes et enregistrements
- Gestion des champs et des liens
- Gestion des tables
- Gestion des requêtes
- Gestion des enregistrements
- Gestion des transactions
- Accès direct aux données
- Gestion du support
- Gestion des logiciels
- Gestion des achats
- Conversion
- Finance
- Exécution des actions
- Calcul des dates et des heures
- Calcul
- Manipulation des chaînes de caractères
- Gestion des fichiers
- Gestion des erreurs
- Fonctions dédiées aux assistants
- Fonctions dédiées au Web
- Diverses

Conventions

Ce chapitre décrit :

- *Conventions d'écriture*
- *Format des constantes de type "Date+Heure" dans les scripts*
- *Format des constantes de type "Durée"*

Conventions d'écriture

La syntaxe des fonctions et des exemples proposés respecte les conventions d'écriture suivantes :

[]	Ces crochets encadrent un paramètre optionnel. Ne les tapez pas dans votre commande. Exception : dans les scripts Basic, lorsque les crochets encadrent le chemin d'accès à des données de la base, ils doivent figurer dans le script, comme le montre l'exemple ci-dessous : `[Lien.Lien.Champ]`
<>	Ces crochets encadrent un paramètre décrit en langage clair. Ne les tapez pas dans votre commande et remplacez le texte qu'ils encadrent par l'information qui doit y figurer.
{}	Ces accolades encadrent des paramètres parmi lesquels un seul doit être choisi. Ne tapez pas les accolades dans votre commande.
|	La barre verticale sépare les paramètres possibles qui figurent dans les accolades.
*	L'astérisque ajouté à droite de crochets indique que la formule qu'ils encadrent peut être répétée plusieurs fois.

Les mises en forme suivantes ont des significations particulières :

`Police fixe`	Commande DOS, paramètre de fonction ou formatage de données.
`Exemple`	Exemple de code ou de commande.
...	Portion de code ou de commande omise.
Nom d'objet	Les noms de champs, d'onglets, de menus, de fichiers sont en caractères gras.

Note: Note	Note importante.

Format des constantes de type "Date+Heure" dans les scripts

Les dates référencées dans les scripts sont exprimées au format international, indépendamment des options d'affichage spécifiées par l'utilisateur :

```
yyyy/mm/dd hh:mm:ss
```

Exemple :

```
RetVal="1998/07/12 13:05:00"
```

Note: Le tiret ("-") peut également être utilisé comme séparateur de date.

Date Basic et Date Unix

Les dates sont exprimées différemment en Basic et Unix :

- En Basic, une date peut être exprimée au format international ou sous la forme d'un nombre à virgule flottante (type "Double"). Dans ce dernier cas, la partie entière de ce nombre représente le nombre de jours écoulés depuis le 30/12/1899 à 0:00, la partie décimale représente la fraction écoulée dans le jour courant.
- Sous Unix, les dates sont exprimées sous la forme d'un entier long (type "Long" 32 bits) qui représente le nombre de secondes écoulées depuis le 01/01/1970 à 0:00, indépendamment d'un quelconque fuseau horaire (heure UTC).

Note: Lorsqu'elles sont appelées par un programme externe, toutes les fonctions de manipulation de dates attendent et retournent des dates GMT.

Format des constantes de type "Durée"

Dans les scripts, les durées sont stockées et exprimées en secondes. Par exemple, pour fixer la valeur par défaut d'un champ de type "Durée" à 3 jours, vous devez utiliser le script suivant :

```
RetVal=259200
```

De même, les fonctions qui calculent une durée, comme la fonction "AmWorkTimeSpanBetween()", fournissent un résultat en secondes.

Note: Dans les calculs financiers, AssetCenter tient compte des simplifications habituellement usitées. Dans ce cas seulement, une année vaut 12 mois et un mois vaut 30 jours (d'où : 1 année = 360 jours).

Définitions

Ce chapitre regroupe les définitions de quelques termes essentiels.

Vous y trouverez les définitions suivantes :

- *Définition d'une fonction*
- *Définition du lien virtuel "CurrentUser"*
- *Définition d'un handle*
- *Définition d'un code d'erreur*

Définition d'une fonction

Une fonction est un programme qui effectue des opérations et renvoie à l'utilisateur une valeur, appelée "valeur de retour" ou "code de retour".

Voici un exemple de syntaxe d'appel d'une fonction par le Basic interne d'AssetCenter :

```
AmConvertCurrency(strSrcName As String, strDstName As String, dVal As
Double) As Double
```

Voici à présent la syntaxe d'appel de la même fonction au travers des APIs AssetCenter :

```
double AmConvertCurrency(long hApiCnxBase,long ltm, const char
*pszSrcName, const char *pszDstName,double dVal)
```

Définition du lien virtuel "CurrentUser"

Définition

"CurrentUser" peut être considéré comme un lien qui part de toutes les tables et pointe vers l'enregistrement de la table des services et personnes correspondant à l'utilisateur courant.

- Sous la forme "CurrentUser", il pointe sur l'enregistrement correspondant à l'utilisateur courant et renvoie son numéro d'identifiant (son "Id").
- Sous la forme "CurrentUser.Champ", il renvoie la valeur du champ pour l'utilisateur courant.

Note: Ce lien virtuel n'est pas affiché dans la liste des champs et des liens et n'est donc pas accessible directement dans le constructeur de scripts interne à AssetCenter. Vous devez saisir cette expression à la main.

Equivalences

Les fonctions "AmLoginName()" et "AmLoginId()" qui fournissent respectivement le "Nom" (Nom SQL : Name) et le numéro d'identifiant (Nom SQL : lPersId) de l'utilisateur courant peuvent être considérées comme des fonctions dérivées de "CurrentUser". En effet, on a les équivalences suivantes :

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.lPersId]

Définition d'un handle

Un handle représente un identifiant unique sur un objet. Dans le contexte d'AssetCenter, cet objet peut être un champ, un lien, une requête, un enregistrement, une table ou une connexion. Les handles sont des entiers (type "Long") codés sur 32 bits.

Note: Un handle valide ne peut avoir une valeur nulle (NULL).

Définition d'un code d'erreur

Lorsque l'exécution d'une fonction échoue, un code d'erreur est renvoyé.

Ce code d'erreur et le message qui lui est associé peuvent être récupérés respectivement grâce aux fonctions "AmLastError()" et "AmLastErrorMsg()". Il peut être détruit au moyen de la fonction "AmClearLastError()".

Note: Tout nouvel appel de fonction efface le code d'erreur et le message précédents.

Vous pouvez déclencher volontairement un message d'erreur en utilisant la fonction `Err.Raise` dont la syntaxe est la suivante :

```
Err.Raise (<Numéro de l'erreur>, <Message d'erreur>)
```

Note: Lorsque la création ou la modification d'un enregistrement est invalidé par la valeur du champ "Validité" pour la table concernée, il est judicieux de déclencher un message d'erreur au moyen de la fonction `Err.Raise` afin de prévenir l'utilisateur. Si vous ne le faites pas, celui-ci ne comprendra pas nécessairement pourquoi il ne peut ni modifier, ni créer l'enregistrement.

Typage des fonctions et des paramètres de fonctions

Vous trouverez dans ce chapitre les informations suivantes :

- *Liste des types*
- *Type d'une fonction*
- *Type d'un paramètre*

Liste des types

Le tableau ci-dessous récapitule les différents types possibles pour une fonction ou un paramètre :

Type	Signification
Integer	Nombre entier de -32 768 à +32 767.
Long	Nombre entier de -2 147 483 647 à +2 147 483 646.

Type	Signification
Double	Nombre à virgule flottante de 8 octets.
String	Texte pour lequel tous les caractères sont acceptés.
Date	Date ou Date+Heure.
Variant	Type générique pouvant représenter n'importe quel type.

Type d'une fonction

Le type d'une fonction correspond au type de la valeur retournée par la fonction. Nous vous invitons à faire particulièrement attention à cette information car elle peut être à l'origine d'erreurs de compilation et d'exécution de vos programmes.

Par exemple, vous ne pouvez pas utiliser une fonction renvoyant une valeur d'un certain type dans la définition de la valeur par défaut d'un champ d'un type différent. Essayez par exemple d'affecter ce script de valeur par défaut à n'importe quel champ de type "Date" ou "Date+Heure" :

```
RetVal=AmLoginName()
```

La fonction "AmLoginName()" renvoie le nom de l'utilisateur connecté sous la forme d'une chaîne de caractères (type "String"). Cette valeur de retour est donc dans un format incompatible avec celui d'un champ de type "Date" et AssetCenter affiche un message d'erreur.

Type d'un paramètre

Les paramètres utilisés dans les fonctions possèdent également un type que vous devez impérativement respecter pour la bonne exécution de la fonction. Dans la syntaxe des fonctions, les paramètres sont préfixés en fonction de leur type. Pour éviter toute confusion possible, les préfixes utilisés dans cette référence sont différents suivant la syntaxe (API ou Basic) de la fonction. Le tableau ci-dessous propose pour chaque type une équivalence entre le préfixe utilisé dans la syntaxe API et celui utilisé dans la syntaxe Basic :

Type	Préfixe utilisé dans la syntaxe API	Préfixe utilisé dans la syntaxe Basic
Integer	"i"	"i"
Long	"h" pour un handle ou "l" pour un nombre	"l"
Double	"d"	"d"
String	"char*psz"	"str"
Date	"ltm"	"dt"
Variant	"v"	"v"

Chapitre 2 - Utilisation des APIs

Ce chapitre propose un rapide tour d'horizon sur l'utilisation des APIs. Il s'articule en trois grands points :

- *Préambule*
- *Méthodologie*
- *Concepts et exemples*

Préambule

Les APIs AssetCenter sont fournies sous la forme d'un fichier DLL 32 bits utilisable sous Windows 95/98 ou Windows NT.

Elles ont été testées avec succès dans des environnements de développement suivants :

- Visual Basic 4.0, 5.0 et 6.0,
- Visual C++ 4.0, 5.0 et 6.0,
- Tous les produits de la gamme Microsoft™ utilisant le VBA (Visual Basic for Applications)

Note: Les APIs sont à priori compatibles avec tous les outils autorisant l'utilisation de bibliothèques de fonctions externes (DLL).

Avertissement

L'utilisation des APIs AssetCenter est assujettie à une bonne connaissance du modèle conceptuel de données utilisé par AssetCenter en général et de la structure de la base de données en particulier.

Toutes les informations utiles sur la structure de la base de données sont regroupées dans le manuel intitulé "Manuel de référence :

Administration et utilisation avancée", chapitre "Structure de la base de données AssetCenter" ainsi que dans les fichiers "database.txt" et "tables.txt", situés dans le sous-répertoire "infos" du répertoire d'installation d'AssetCenter.

Installation

Avant toute utilisation des APIs, nous vous recommandons de procéder à une installation complète d'AssetCenter. Vous pourrez ainsi contrôler facilement l'accès aux bases de données à partir de votre ordinateur, créer et configurer rapidement les bases de données sur lesquelles vous souhaitez travailler. Les APIs utilisant les mêmes couches de base de données et les mêmes informations de configuration qu'AssetCenter pour l'accès aux sources de données, vous pouvez rapidement détecter, à partir de l'interface graphique d'AssetCenter, les problèmes rencontrés lors de l'utilisation des APIS.

Les étapes classiques pour l'installation d'un environnement de développement sous AssetCenter sont les suivantes :

- Installation d'une version 32 bits d'AssetCenter avec le composant AssetCenter APIs.
- Configuration de la source de données et test d'accès à la base de données sous AssetCenter.
- Utilisation de votre environnement de développement pour appeler les fonctions des APIs AssetCenter.

Nous vous recommandons de vous familiariser avec les APIs AssetCenter en utilisant une base de données de démonstration ou toute autre source ne contenant aucune donnée sensible.

Méthodologie

Voici, étape par étape, une approche classique de l'utilisation des APIs AssetCenter :

1. Création d'une requête AQL du type :

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amAsset
```

Note: Une bonne solution pour composer simplement une requête AQL est d'utiliser AssetCenter Export.

2. Récupération des résultats de la requête et des handles sur tous les objets que vous souhaitez utiliser.

3. Utilisation des ces handles pour mettre à jour les informations contenues dans les objets correspondants.

4. Réalisation d'un "Commit" (pour accepter toutes les modifications) ou d'un "Rollback" (pour annuler toutes les modifications) pour la transaction en cours.

Concepts et exemples

Vous trouverez dans cette partie les informations suivantes :

- *Concepts*
- *Manipuler les dates*
- *Premier exemple*
- *Second exemple*

Concepts

AssetCenter a été pensé et conçu dans une approche "objet" qui se retrouve également dans les APIs. Les DLLs de Windows requièrent l'utilisation d'un modèle "flat" proche du C. Les APIs AssetCenter contournent cette limitation en utilisant des "handles" sur les objets créés par l'utilisateur. Cette approche permet aux langages qui ne sont pas orientés "objet" d'accéder au modèle d'AssetCenter.

Avant tout autre chose, votre programme doit utiliser la fonction "AmStartUp()" pour initialiser l'appel aux bibliothèques AssetCenter. De même, la dernière fonction appelée par votre programme doit impérativement être la fonction "AmCleanUp()".

Avant tout accès à un objet d'une base, une connexion valide doit être établie entre l'utilisateur et cette base de données. Cette connexion est identifiée par un "handle" sur un objet "connexion" (ce handle est alors utilisé dans toutes les fonctions des APIs qui interagissent avec la base de données. Il correspond au paramètre "hApiCnxBase"). Cet objet peut alors être utilisé pour créer des requêtes et accéder aux enregistrements.

Note: Notez que tous les objets d'une base de données sont liés à une connexion. Il en résulte que les informations sur les droits d'accès par exemple peuvent être contrôlées.

La première étape consiste donc en la création d'une connexion utilisant une source de données, un login et son mot de passe valides.

Avertissement

Attention : lorsque vous vous connectez à une base de données AssetCenter au travers des APIs, un jeton de connexion est utilisé.

Manipuler les dates

Pour lire une date, vous avez la possibilité d'utiliser l'une des deux fonctions suivantes sur un champ de type "Date" ou "Date+Heure" :

- "AmGetFieldLongValue()" qui renvoie la date sous la forme d'un "Long" Unix (UTC). Préférez cette fonction pour les calculs faisant intervenir les dates.
- "AmGetFieldStrValue()" qui renvoie la date sous la même forme que le "control panel" de Windows. Cette date respecte les fuseaux horaires. Utilisez cette fonction pour l'affichage.

Premier exemple

L'exemple suivant, rédigé en C, déclare une connexion à la base de démonstration :

```
long lCnx ;

lCnx = AmOpenConnection("ACDemo300FRA","Admin" ,"") ;
```

"lCnx" représente un "handle" sur un objet "connexion". Ce handle est utilisé pour identifier la connexion que vous venez de déclarer.

Cette connexion peut dès lors être utilisée pour créer des requêtes et accéder à la base de données. L'exemple suivant, rédigé en C, définit une requête sur la table des biens et parcourt les résultats de cette requête :

```
#include "apiproto.h"
#define SZ_MODEL_LEN 200
long lCnx ;
long lQuery ;
long lStatus ;  /* to store error code */
char szModel[SZ_MODEL_LEN] ;
/* dll initialization */
```

```
AmStartup();
/* Open a connection */
lCnx = AmOpenConnection("ACDemo300Eng","Admin" ,"") ;
if( lCnx != 0 )
{
  /* Creation of a query object */
  lQuery = AmQueryCreate (lCnx)
  if( lQuery != 0 )
  {
    /* Construction of  the result set : all assets from Compaq*/
  lStatus = AmQueryExec(lQuery, "select AssetTag where brand =
'Compaq'")
    /* Navigates through the result set */
    while( !lStatus )
    {
        /* Read the first field (AssetTag) of the current item in the
query */
        lStatus = AmGetFieldStrValue(lQuery,0,szModel,SZ_MODEL_LEN-1);
        if( lStatus == 0 )
        {
        printf(' Compaq AssetTag=%s\n',szModel);
        lStatus = AmQueryNext(lQuery);
      }
    }
    /* clean things up */
    AmReleaseHandle(lQuery);
  }
  AmCloseConnection(lCnx);
}
AmCleanup();
```

Second exemple

Les requêtes sont utilisées pour localiser les objets dans la base de données. Lorsque vous devez mettre à jour un enregistrement, un handle sur l'objet "enregistrement" doit être récupéré au moyen d'une requête. L'enregistrement peut alors être traité au moyen des autres fonctions des APIs AssetCenter.

L'exemple suivant illustre la modification d'un champ d'un enregistrement donné :

```
/* Handles for objects */
long lCnx ;
long lQuery ;
long lStatus ;
long lRecord ;
AmStartup();
lCnx = AmOpenConnection("ACDemo300Eng","Admin" ,"") ;
/* Creation of a query object attached to lCnx */
lQuery = AmQueryCreate(lCnx);
/* Mark the starting point of the current transaction */
AmStartTransaction(lCnx);
```

```
/* Use a query that matches a single object */
lStatus = AmQueryGet(lQuery, "select model, AssetId where brand =
'Compaq' and barcode='34234'") ;
/* Get a record handle to the matching object */
lRecord = AmGetRecordHandle(lQuery) ;
/* Change the field Field1 with new value spam */
lStatus = AmSetFieldStrValue(lRecord, "Field1", "Spam");
/* Update the change for the current session */
lStatus = AmUpdateRecord(lrecord);
/* Commit all modifications to the database */
lStatus = AmCommit(lCnx) ;
/* you can release here query and record objects */
/* but closing connection will do it */
/* Close the connection to the database */
AmCloseConnection(lCnx);
AmCleanup();
```

Cet exemple illustre la récupération d'un "handle" sur un enregistrement par le biais d'une requête. La requête est alors analysée pour localiser un élément de l'enregistrement (dans cet exemple un champ). Il est également possible d'utiliser la fonction "AmQueryExec()" pour récupérer plusieurs enregistrements en une seule requête, puis de parcourir ces enregistrements et récupérer les "handle" des éléments qui vous intéressent.

Note: Par souci de simplification, cet exemple ne traite pas tous les codes d'erreur possibles.

Chapitre 3 - Référence alphabétique

Abs()

Syntaxe Basic interne

```
Function Abs(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	X

Description

Renvoie la valeur absolue d'un nombre.

Entrée

- *dValue* : Nombre dont vous souhaitez connaître la valeur absolue.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main
    Dim Msg, X, Y
    X = InputBox("Enter a Number:")
    Y = Abs(X)
    Msg = "The number you entered is "  & X
    Msg = Msg + ". The Absolute value of " & X & " is " & Y
    Print Msg 'Display Message.
End Sub
```

AmActionDde()

Syntaxe API

```
long AmActionDde(char *strService, char *strTopic, char *strCommand,
char *strFileName, char *strDirectory, char *strParameters, char
*strTable, long lRecordId);
```

Syntaxe Basic interne

```
Function AmActionDde(strService As String, strTopic As String,
strCommand As String, strFileName As String, strDirectory As String,
strParameters As String, strTable As String, lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction lance une requête DDE à destination d'une application qui gère les liens DDE. Grâce à cette fonction, AssetCenter peut piloter une autre application par l'intermédiaire d'un lien DDE.

Entrée

- *strService* : Ce paramètre contient le nom du service DDE proposé par l'exécutable que vous souhaitez solliciter. Veuillez vous reporter à la documentation de cet exécutable pour connaître la liste des services DDE qu'il propose.
- *strTopic* : Ce paramètre contient le thème, c'est-à-dire le contexte dans lequel l'action DDE doit être effectuée.
- *strCommand* : Ce paramètre contient les commandes que l'application externe doit exécuter. Vous devez respecter la syntaxe imposée par l'application externe.
- *strFileName* : Si le service n'est pas présent en mémoire, vous devez le charger en précisant dans ce paramètre le nom de l'exécutable (ou celui d'un fichier quelconque si celui-ci est associé à un exécutable par l'intermédiaire du gestionnaire de fichiers de Windows) qui active le service.
- *strDirectory* : Ce paramètre contient le chemin d'accès du fichier précisé dans *strFileName*.
- *strParameters* : Ce paramètre contient les différents paramètres à fournir à l'exécutable qui active le service lors de son lancement.
- *strTable* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique le nom SQL de la table contenant l'enregistrement auquel s'applique l'action.
- *lRecordId* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique l'identifiant de l'enregistrement auquel s'applique l'action.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmActionExec()

Syntaxe API

```
long AmActionExec(char *strFileName, char *strDirectory, char
*strParameters, char *strTable, long lRecordId);
```

Syntaxe Basic interne

```
Function AmActionExec(strFileName As String, strDirectory As String,
strParameters As String, strTable As String, lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction lance une application de type ".exe", ".com", ".bat", ".pif". Vous pouvez également faire référence à des documents de tous types, à

condition que leur extension soit associée à un exécutable par le gestionnaire de fichiers de Windows. Cette fonction est équivalente à une action de type "Exécutable".

Entrée

- *strFileName* : Ce paramètre contient le nom de l'exécutable ou celui d'un document quelconque (associé à un exécutable par l'intermédiaire du gestionnaire de fichiers de Windows).
- *strDirectory* : Ce paramètre contient le chemin d'accès du fichier précisé dans le paramètre *strFileName*.
- *strParameters* : Ce paramètre optionnel contient les différents paramètres à fournir à l'exécutable lors de son lancement.
- *strTable* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique le nom SQL de la table contenant l'enregistrement auquel s'applique l'action.
- *lRecordId* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique l'identifiant de l'enregistrement auquel s'applique l'action.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

Cet exemple exécute l'explorer de Windows NT (situé dans le dossier "WinNT" du disque "C") :

```
AmActionExec(explorer.exe, c:\winnt\, )
```

AmActionMail()

Syntaxe API

```
long AmActionMail(char *strTo, char *strCc, char *strCcc, char
*strSubject, char *strMessage, long iPriority, long bAcknowledge, char
*strRefObject, char *strTable, long lRecordId);
```

Syntaxe Basic interne

```
Function AmActionMail(strTo As String, strCc As String, strCcc As
String, strSubject As String, strMessage As String, iPriority As Long,
bAcknowledge As Long, strRefObject As String, strTable As String,
lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction émet un message via l'un des messageries gérées par AssetCenter :

- Messagerie interne.
- Messagerie externe au standard VIM (Lotus Notes, ...).
- Messagerie externe au standard MAPI (Microsoft Exchange, Microsoft Outlook, ...).
- Messagerie externe au standard SMTP (standard Internet).

Entrée

- *strTo* : Ce paramètre contient la liste des adresses des destinataires du message. Le point-virgule est utilisée comme séparateur.
- *strCc* : Ce paramètre contient la liste des adresses des destinataires en copie du message. Le point-virgule est utilisée comme séparateur.
- *strCcc* : Ce paramètre contient la liste des adresses des destinataires en copie cachée du message (ils n'apparaissent pas dans la liste des destinataires). Le point-virgule est utilisée comme séparateur.
- *strSubject* : Ce paramètre contient l'intitulé du message.
- *strMessage* : Ce paramètre contient le corps du message.
- *iPriority* : Ce paramètre définit la priorité d'envoi du message
  - √ 0 priorité basse.
  - √ 1 priorité normale.
  - √ 2 priorité haute.
- *bAcknowledge* : Ce paramètre précise si l'émetteur du message reçoit un accusé de réception :
  - √ 0 : l'émetteur ne reçoit pas d'accusé de réception.
  - √ 1 l'émetteur reçoit un accusé de réception.
- *strRefObject* : Ce paramètre ne sert qu'aux messages adressés à la messagerie interne d'AssetCenter. Il permet d'accéder directement à l'objet qui a déclenché l'émission du message.
- *strTable* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique le nom SQL de la table contenant l'enregistrement auquel s'applique l'action.

- *lRecordId* : Paramètre optionnel, utilisé dans le cas où l'action est contextuelle. Il indique l'identifiant de l'enregistrement auquel s'applique l'action.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmActionPrint()

Syntaxe API

```
long AmActionPrint(long lReportId, long lRecordId);
```

Syntaxe Basic interne

```
Function AmActionPrint(lReportId As Long, lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction déclenche l'impression d'un rapport sur un enregistrement donné de la base.

Entrée

- *lReportId* : Ce paramètre contient l'identifiant du rapport à imprimer.
- *lRecordId* : Ce paramètre contient l'identifiant de l'enregistrement concerné par le rapport. Par défaut, ce paramètre prend la valeur "0".

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddAllPOLinesToInv()

Syntaxe API

```
long AmAddAllPOLinesToInv(long hApiCnxBase, long lPOrdId, long lInvId);
```

Syntaxe Basic interne

```
Function AmAddAllPOLinesToInv(lPOrdId As Long, lInvId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute l'intégralité d'une commande à une facture fournisseur existante.

Entrée

- `lPOrdId` : Ce paramètre contient l'identifiant de commande à ajouter à la facture fournisseur.
- `lInvId` : Ce paramètre contient l'identifiant de la facture fournisseur à laquelle est ajoutée la commande.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddEstimLinesToPO()

Syntaxe API

```
long AmAddEstimLinesToPO(long hApiCnxBase, long lEstimId, long lPOrdId,
long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddEstimLinesToPO(lEstimId As Long, lPOrdId As Long,
bMergeLines As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute toutes les lignes de devis d'un devis à une commande existante.

Entrée

- *lEstimId* : Ce paramètre contient l'identifiant du devis à ajouter à la commande.
- *lPOrdId* : Ce paramètre contient l'identifiant de la commande à laquelle sont ajoutées toutes les lignes de devis du devis.
- *bMergeLines* : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (*bMergeLines*=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddEstimLineToPO()

Syntaxe API

```
long AmAddEstimLineToPO(long hApiCnxBase, long lEstimLineId, long
lPOrdId, long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddEstimLineToPO(lEstimLineId As Long, lPOrdId As Long,
bMergeLines As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X

	Utilisable
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute une ligne de devis à une commande existante.

Entrée

- `lEstimLineId` : Ce paramètre contient l'identifiant de la ligne de devis à ajouter à la commande.
- `lPOrdId` : Ce paramètre contient l'identifiant de la commande à laquelle est ajoutée la ligne de devis.
- `bMergeLines` : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (`bMergeLines`=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddPOLineToInv()

Syntaxe API

```
long AmAddPOLineToInv(long hApiCnxBase, long lPOrdLineId, long lInvId,
long lQty);
```

Syntaxe Basic interne

```
Function AmAddPOLineToInv(lPOrdLineId As Long, lInvId As Long, lQty As
Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute une quantité donnée d'élément(s) sur une ligne de commande à une facture fournisseur.

Entrée

- *lPOrdLineId* : Ce paramètre contient l'identifiant de la ligne de commande à ajouter à la facture fournisseur.
- *lInvId* : Ce paramètre contient l'identifiant de la facture fournisseur à laquelle des éléments de la ligne de commande sont ajoutés.
- *lQty* : Ce paramètre contient la quantité d'éléments présents sur la ligne de commande à ajouter à la facture fournisseur.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmAddReqLinesToEstim()

Syntaxe API

```
long AmAddReqLinesToEstim(long hApiCnxBase, long lReqId, long lEstimId,
long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddReqLinesToEstim(lReqId As Long, lEstimId As Long,
bMergeLines As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute toutes les lignes de demande d'une demande à un devis existant.

Entrée

- `lReqId` : Ce paramètre contient l'identifiant de la demande à ajouter au devis.
- `lEstimId` : Ce paramètre contient l'identifiant du devis auquel sont ajoutées toutes les lignes de demande de la demande.
- `bMergeLines` : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (`bMergeLines`=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddReqLinesToPO()

Syntaxe API

```
long AmAddReqLinesToPO(long hApiCnxBase, long lReqId, long lPOrdId,
long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddReqLinesToPO(lReqId As Long, lPOrdId As Long, bMergeLines
As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute toutes les lignes de demande d'une demande à une commande existante. Le fournisseur précisé dans la demande doit être identique à celui de la commande concernée.

Entrée

- *lReqId* : Ce paramètre contient l'identifiant de la demande à ajouter à la commande.
- *lPOrdId* : Ce paramètre contient l'identifiant de la commande à laquelle sont ajoutées les lignes de demande de la demande.
- *bMergeLines* : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (*bMergeLines*=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddReqLineToEstim()

Syntaxe API

```
long AmAddReqLineToEstim(long hApiCnxBase, long lReqLineId, long
lEstimId, long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddReqLineToEstim(lReqLineId As Long, lEstimId As Long,
bMergeLines As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute une ligne de demande à un devis existant.

Entrée

- `lReqLineId` : Ce paramètre contient l'identifiant de la ligne de demande à ajouter au devis.
- `lEstimId` : Ce paramètre contient l'identifiant du devis auquel est ajoutée la ligne de demande.
- `bMergeLines` : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (`bMergeLines`=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmAddReqLineToPO()

Syntaxe API

```
long AmAddReqLineToPO(long hApiCnxBase, long lReqLineId, long lPOrdId,
long bMergeLines);
```

Syntaxe Basic interne

```
Function AmAddReqLineToPO(lReqLineId As Long, lPOrdId As Long,
bMergeLines As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction ajoute une ligne de demande à une commande existante.

Entrée

- *lReqLineId* : Ce paramètre contient l'identifiant de la ligne de demande à ajouter à la commande.
- *lPOrdId* : Ce paramètre contient l'identifiant de la commande à laquelle est ajoutée la ligne de demande.
- *bMergeLines* : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (*bMergeLines*=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmApproveRequest()

Syntaxe API

```
long AmApproveRequest(long hApiCnxBase, long lReqId, char *strComment,
long bCheckCompo, long bCheckCost, double dMaxCost);
```

Syntaxe Basic interne

```
Function AmApproveRequest(lReqId As Long, strComment As String,
bCheckCompo As Long, bCheckCost As Long, dMaxCost As Double) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmBusinessSecondsInDay()

Syntaxe API

```
long AmBusinessSecondsInDay(char *strCalendarSqlName, long tmDate);
```

Syntaxe Basic interne

```
Function AmBusinessSecondsInDay(strCalendarSqlName As String, tmDate As
Date) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Calcule le nombre de secondes ouvrées dans une journée en fonction d'un calendrier.

Entrée

- `strCalendarSqlName` : Nom SQL du calendrier utilisé pour le calcul.
- `tmDate` : Date à laquelle s'effectue le calcul.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCalcConsolidatedFeature()

Syntaxe API

```
long AmCalcConsolidatedFeature(long hApiCnxBase, long lCalcFeatId, char
*strSQLTableName);
```

Syntaxe Basic interne

```
Function AmCalcConsolidatedFeature(lCalcFeatId As Long, strSQLTableName
As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Calcule la valeur d'une caractéristique consolidée sur une table identifiée par son nom SQL.

Entrée

- *lCalcFeatId* : Identifiant de la caractéristique consolidée.
- *strSQLTableName* : Nom SQL de la table pour laquelle la caractéristique consolidée est calculée. La caractéristique doit absolument être définie pour cette table.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCalcDepr()

Syntaxe API

```
double AmCalcDepr(long iType, long lDuration, double dCoeff, double
dPrice, long tmStart, long tmDate);
```

Syntaxe Basic interne

```
Function AmCalcDepr(iType As Long, lDuration As Long, dCoeff As Double,
dPrice As Double, tmStart As Date, tmDate As Date) As Double
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction permet de calculer le montant de l'amortissement sur un bien à une date donnée. Elle renvoie la valeur de l'amortissement à cette date.

Entrée

- `iType` : Ce paramètre permet d'identifier la nature de l'amortissement. Les valeurs possibles de ce paramètre sont les suivantes :
  - √ 0 : pas d'amortissement
  - √ 1 : amortissement linéaire
  - √ 2 : amortissement dégressif
- `lDuration` : Ce paramètre contient la durée sur laquelle porte l'amortissement du bien. Cette durée est exprimée en secondes.
- `dCoeff` : Ce paramètre contient le coefficient appliqué lors du calcul de l'amortissement dégressif. Il n'est pas interprété dans le cas d'un amortissement linéaire mais doit posséder une valeur quelconque.
- `dPrice` : Ce paramètre contient la valeur initiale du bien sur lequel porte le calcul de l'amortissement.

- *tmStart* : Ce paramètre contient la date à partir de laquelle le bien est amorti.
- *tmDate* : Ce paramètre contient la date à laquelle sont évalués l'amortissement et la valeur résiduelle du bien.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCleanup()

Syntaxe Basic interne

```
Champ d'application
```

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	X

Description

Cette fonction doit être appelée à la fin de tout script utilisant les fonctions de modification de la base de données. Elle libère toutes les ressources utilisées.

AmClearLastError()

Syntaxe API

```
long AmClearLastError(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmClearLastError() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction efface les informations concernant le dernier message d'erreur survenu lors de la connexion.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCloseAllChildren()

Syntaxe API

```
long AmCloseAllChildren(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmCloseAllChildren() As Long
```

Champ d'application

Version : 2.52

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction détruit tous les objets créés lors de la connexion courante.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCloseConnection()

Syntaxe API

```
long AmCloseConnection(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmCloseConnection() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Met un terme à la session AssetCenter pour une connexion donnée. Tous les objets (requêtes, enregistrements, tables, champs, …) créés au cours de la session sont automatiquement détruits et tous leurs handles deviennent obsolètes et sont inutilisables.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCommit()

Syntaxe API

```
long AmCommit(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmCommit() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction réalise un "Commit" de toutes les modifications apportées à la base de données associée à la connexion.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmComputeAllLicAndInstallCounts()

Syntaxe API

```
long AmComputeAllLicAndInstallCounts(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmComputeAllLicAndInstallCounts() As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue le décompte des licences et des installations logicielles pour tous les enregistrements.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmComputeLicAndInstallCounts()

Syntaxe API

```
long AmComputeLicAndInstallCounts(long hApiCnxBase, long lSLCountId);
```

Syntaxe Basic interne

```
Function AmComputeLicAndInstallCounts(lSLCountId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue le décompte des licences et des installations logicielles pour un enregistrement.

Entrée

- *`lSLCountId`* : Ce paramètre contient l'identifiant du compteur de licences logicielles.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmConvertCurrency()

Syntaxe API

```
double AmConvertCurrency(long hApiCnxBase, long tmDate, char
*strSrcName, char *strDstName, double dVal);
```

Syntaxe Basic interne

```
Function AmConvertCurrency(tmDate As Date, strSrcName As String,
strDstName As String, dVal As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction effectue la conversion entre deux devises à une date et avec une précision données.

Entrée

- `tmDate` : Ce paramètre contient la date de conversion. Elle permet de connaître le taux de conversion effectif à cette date.
- `strSrcName` : Ce paramètre contient la devise source de la conversion, c'est-à-dire celle que vous souhaitez convertir.
- `strDstName` : Ce paramètre contient la devise de destination de la conversion, c'est-à-dire celle dans laquelle sera exprimée la devise source.
- `dVal` : Ce paramètre contient le montant (exprimé dans l'unité monétaire de la devise source) à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Les devises utilisées comme paramètres pour la fonction (*`strSrcName`* et *`strDstName`*) doivent impérativement être définies sous AssetCenter. De même un taux de change valide doit exister à la date à laquelle s'effectue la conversion (paramètre *`tmDate`*).

Exemple

L'exemple suivant effectue la conversion de 5000 FRF en dollars, à la date du 02/11/98.

```
AmConvertCurrency("1998/11/02 00:00:00", "FRF", "$", 5000)
```

AmConvertDateBasicToUnix()

Syntaxe API

```
long AmConvertDateBasicToUnix(long hApiCnxBase, long tmTime);
```

Syntaxe Basic interne

```
Function AmConvertDateBasicToUnix(tmTime As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format Basic (type "Date") en une date au format Unix (type "Long").

Entrée

- *`tmTime`* : Ce paramètre contient la date à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDateIntlToUnix()

Syntaxe API

```
long AmConvertDateIntlToUnix(long hApiCnxBase, char *strDate);
```

Syntaxe Basic interne

```
Function AmConvertDateIntlToUnix(strDate As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format international (type "Date") en une date au format Unix (type "Long").

Entrée

- *strDate* : Ce paramètre contient la date à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDateStringToUnix()

Syntaxe API

```
long AmConvertDateStringToUnix(long hApiCnxBase, char *strDate);
```

Syntaxe Basic interne

```
Function AmConvertDateStringToUnix(strDate As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de	X

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une date au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows) en un "Long" Unix.

Entrée

- *`strDate`* : Date au format chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDateUnixToBasic()

Syntaxe API

```
long AmConvertDateUnixToBasic(long hApiCnxBase, long lTime);
```

Syntaxe Basic interne

```
Function AmConvertDateUnixToBasic(lTime As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format Unix (type "Long") en une date au format Basic (type "Date")

Entrée

- `lTime` : Ce paramètre contient la date à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDateUnixToIntl()

Syntaxe API

```
long AmConvertDateUnixToIntl(long hApiCnxBase, long lUnixDate, char
*pstrDate, long lDate);
```

Syntaxe Basic interne

```
Function AmConvertDateUnixToIntl(lUnixDate As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format Unix (type "Long") en une date au format international (type "Long").

Entrée

- *lUnixDate* : Ce paramètre contient la date à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDateUnixToString()

Syntaxe API

```
long AmConvertDateUnixToString(long hApiCnxBase, long lUnixDate, char
*pstrDate, long lDate);
```

Syntaxe Basic interne

```
Function AmConvertDateUnixToString(lUnixDate As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une date au format "Long" Unix en une date au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows).

Entrée

- `lUnixDate` : Date au format "Long" Unix à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertDoubleToString()

Syntaxe API

```
long AmConvertDoubleToString(double dSrc, char *pstrDst, long lDst);
```

Syntaxe Basic interne

```
Function AmConvertDoubleToString(dSrc As Double) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit un nombre en double précision en une chaîne de caractères.

Entrée

- *dSrc* : Ce paramètre contient le nombre en double précision à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertMonetaryToString()

Syntaxe API

```
long AmConvertMonetaryToString(double dSrc, char *pstrDst, long lDst);
```

Syntaxe Basic interne

```
Function AmConvertMonetaryToString(dSrc As Double) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une valeur monétaire en une chaîne de caractères.

Entrée

- `dSrc` : Ce paramètre contient la valeur monétaire à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertStringToDouble()

Syntaxe API

```
double AmConvertStringToDouble(char *strSrc);
```

Syntaxe Basic interne

```
Function AmConvertStringToDouble(strSrc As String) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une chaîne de caractères en un nombre en double précision.

Entrée

- `strSrc` : Ce paramètre contient la chaîne de caractères à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmConvertStringToMonetary()

Syntaxe API

```
double AmConvertStringToMonetary(char *strSrc);
```

Syntaxe Basic interne

```
Function AmConvertStringToMonetary(strSrc As String) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une chaîne de caractères en une valeur monétaire.

Entrée

- *strSrc* : Ce paramètre contient la chaîne de caractères à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCounter()

Syntaxe API

```
long AmCounter(char *return, long lreturn, char *strCounterName, long
iWidth);
```

Syntaxe Basic interne

```
Function AmCounter(strCounterName As String, iWidth As Long) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction renvoie la valeur du compteur *strCounterName* incrémentée de 1 et exprimée sur *iWidth* chiffres. Des zéros sont ajoutés en préfixe si *iWidth* est supérieur à la valeur du compteur.

Entrée

- *strCounterName* : Nom du compteur tel qu'il est défini sous AssetCenter (accès par le menu Outils/ Administration/ Compteurs).
- *iWidth* : La valeur de ce paramètre force le formatage du résultat de la fonction, en l'exprimant sur n chiffres.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant renvoie la valeur du compteur "BonsLivraison" exprimée sur 5 chiffres :

```
AmCounter("BonsLivraison", 5)
```

Si par exemple le compteur "BonsLivraison" vaut "18", le résultat est le suivant :

```
00019
```

AmCreateDelivFromPO()

Syntaxe API

```
long AmCreateDelivFromPO(long hApiCnxBase, long lPOrdId);
```

Syntaxe Basic interne

```
Function AmCreateDelivFromPO(lPOrdId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la réception d'une commande à partir d'une commande et renvoie l'identifiant de la fiche de réception créée.

Entrée

- *`lPOrdId`* : Ce paramètre contient l'identifiant de la commande à réceptionner.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCreateEstimFromReq()

Syntaxe API

```
long AmCreateEstimFromReq(long hApiCnxBase, long lReqId, long lSuppId);
```

Syntaxe Basic interne

```
Function AmCreateEstimFromReq(lReqId As Long, lSuppId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la création d'une devis à partir d'une demande d'achat et renvoie l'identifiant du devis créé.

Entrée

- `lReqId` : Ce paramètre contient l'identifiant de la demande d'achat qui sert à la création du devis.
- `lSuppId` : Ce paramètre contient l'identifiant du fournisseur du devis qui sera créé à l'issue de l'exécution de la fonction.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCreateEstimsFromAllReqLines()

Syntaxe API

```
long AmCreateEstimsFromAllReqLines(long hApiCnxBase, long lReqId, long
bMergeLines, long lDefSuppId);
```

Syntaxe Basic interne

```
Function AmCreateEstimsFromAllReqLines(lReqId As Long, bMergeLines As
Long, lDefSuppId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction crée un devis à partir d'une demande et renvoie l'identifiant du devis créé.

Entrée

- *lReqId* : Ce paramètre contient l'identifiant de la demande à l'origine du devis.
- *bMergeLines* : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (*bMergeLines*=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.
- *lDefSuppId* : Ce paramètre contient l'identifiant du fournisseur par défaut pour le devis.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCreateInvFromPO()

Syntaxe API

```
long AmCreateInvFromPO(long hApiCnxBase, long lPOrdId);
```

Syntaxe Basic interne

```
Function AmCreateInvFromPO(lPOrdId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction crée une facture fournisseur à partir d'une commande et renvoie l'identifiant de la facture fournisseur créée.

Entrée

- `lPOrdId` : Ce paramètre contient l'identifiant de la commande à l'origine de la facture.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCreateLink()

Syntaxe API

```
long AmCreateLink(long hApiRecord, char *strLinkName, long
hApiRecDest);
```

Syntaxe Basic interne

```
Function AmCreateLink(hApiRecord As Long, strLinkName As String,
hApiRecDest As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction modifie un lien d'un enregistrement et le fait pointer vers un nouvel enregistrement (`hApiRecDest`) dans la table de destination.

Entrée

- *hApiRecord* : Ce paramètre contient le handle sur l'enregistrement contenant le lien à modifier.
- *strLinkName* : Ce paramètre contient le nom SQL du lien à modifier.
- *hApiRecDest* : Ce paramètre contient un handle sur l'enregistrement de destination du lien.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCreatePOFromEstim()

Syntaxe API

```
long AmCreatePOFromEstim(long hApiCnxBase, long lEstimId);
```

Syntaxe Basic interne

```
Function AmCreatePOFromEstim(lEstimId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de	

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la création d'une commande à partir d'un devis et renvoie l'identifiant de la commande créée.

Entrée

- `lEstimId` : Ce paramètre contient l'identifiant du devis qui sert à la création de la commande.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCreatePOFromReq()

Syntaxe API

```
long AmCreatePOFromReq(long hApiCnxBase, long lReqId, long lSuppId);
```

Syntaxe Basic interne

```
Function AmCreatePOFromReq(lReqId As Long, lSuppId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la création d'une commande à partir d'une demande d'achat et renvoie l'identifiant de la commande créée.

Entrée

- `lReqId` : Ce paramètre contient l'identifiant de la demande d'achat qui sert à la création de la commande.
- `lSuppId` : Ce paramètre contient l'identifiant du fournisseur de la commande qui sera créée à l'issue de l'exécution de la fonction.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCreatePOsFromAllReqLines()

Syntaxe API

```
long AmCreatePOsFromAllReqLines(long hApiCnxBase, long lReqId, long
bMergeLines, long lDefSuppId);
```

Syntaxe Basic interne

```
Function AmCreatePOsFromAllReqLines(lReqId As Long, bMergeLines As
Long, lDefSuppId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X

	Utilisable
APIs AssetCenter Web	X

Description

Cette fonction crée toutes les commandes à partir des lignes de demande d'une demande.

Entrée

- *lReqId* : Ce paramètre contient l'identifiant de la demande à partir de laquelle les commandes vont être créées.
- *bMergeLines* : Ce paramètre permet de préciser si les lignes de demande identiques doivent être combinées (*bMergeLines*=0) pour n'en créer qu'une seule. Les quantités décrites sur les lignes à combiner sont alors ajoutées et une seule ligne est créée.
- *lDefSuppId* : Ce paramètre contient l'identifiant du fournisseur par défaut des éléments demandés. Ce paramètre est optionnel et possède "0" comme valeur par défaut..

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmCreateRecord()

Syntaxe API

```
long AmCreateRecord(long hApiCnxBase, char *strTable);
```

Syntaxe Basic interne

```
Function AmCreateRecord(strTable As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction créée un enregistrement vide dans une table en tenant compte des valeurs par défaut. Ce nouvel enregistrement ne possède aucune existence dans la base de données tant qu'il n'a pas été inséré.

Entrée

- `strTable` : Ce paramètre contient le nom SQL de la table dans laquelle vous souhaitez créer l'enregistrement.

AmCurrentDate()

Syntaxe API

```
long AmCurrentDate();
```

Syntaxe Basic interne

```
Function AmCurrentDate() As Date
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la date courante sur le poste client.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCurrentLanguage()

Syntaxe API

```
long AmCurrentLanguage(char *pstrLanguage, long lLanguage);
```

Syntaxe Basic interne

```
Function AmCurrentLanguage() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la langue utilisée dans AssetCenter ("FR" pour le français, "US" pour l'anglais,...).

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmCurrentServerDate()

Syntaxe API

```
long AmCurrentServerDate(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmCurrentServerDate() As Date
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la date courante sur le serveur.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDateAdd()

Syntaxe API

```
long AmDateAdd(long tmStart, long tsDuration);
```

Syntaxe Basic interne

```
Function AmDateAdd(tmStart As Date, tsDuration As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction calcule une nouvelle date en fonction d'une date de départ à laquelle est ajoutée une durée réelle.

Entrée

- *tmStart* : Ce paramètre contient la date à laquelle sera ajoutée une durée.
- *tsDuration* : Ce paramètre contient la durée à ajouter à la date *tmStart*.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDateAddLogical()

Syntaxe API

```
long AmDateAddLogical(long tmStart, long tsDuration);
```

Syntaxe Basic interne

```
Function AmDateAddLogical(tmStart As Date, tsDuration As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction calcule une nouvelle date en fonction d'une date de départ à laquelle est ajoutée une durée logique (un mois comporte 30 jours).

Entrée

- *tmStart* : Ce paramètre contient la date à laquelle sera ajoutée une durée.
- *tsDuration* : Ce paramètre contient la durée à ajouter à la date *tmStart*.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDateDiff()

Syntaxe API

```
long AmDateDiff(long tmEnd, long tmStart);
```

Syntaxe Basic interne

```
Function AmDateDiff(tmEnd As Date, tmStart As Date) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou	X

	Utilisable
d'un lien	
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction calcule en secondes la durée écoulée entre deux dates.

Entrée

- *tmEnd* : Ce paramètre contient la date de fin de la période sur laquelle est effectué le calcul.
- *tmStart* : Ce paramètre contient la date de début de la période sur laquelle est effectué le calcul.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant calcule la durée écoulée entre le 01/01/98 et le 01/01/99.

```
AmDateDiff("1998/01/01 00:00:00", "1999/01/01 00:00:00")
```

AmDbGetDate()

Syntaxe API

```
long AmDbGetDate(long hApiCnxBase, char *strQuery);
```

Syntaxe Basic interne

```
Function AmDbGetDate(strQuery As String) As Date
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le résultat au format date, de l'exécution d'une requête AQL.

Entrée

- *strQuery* : Ce paramètre contient l'intégralité de la requête AQL dont on veut récupérer le résultat.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetDouble()

Syntaxe API

```
double AmDbGetDouble(long hApiCnxBase, char *strQuery);
```

Syntaxe Basic interne

```
Function AmDbGetDouble(strQuery As String) As Double
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le résultat (sous la forme d'un nombre en double précision), de l'exécution d'une requête AQL.

Entrée

- *`strQuery`* : Ce paramètre contient l'intégralité de la requête AQL dont on veut récupérer le résultat.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetList()

Syntaxe API

```
long AmDbGetList(long hApiCnxBase, char *strQuery, char *pstrResult,
long lResult, char *strColSep, char *strLineSep, char *strIdSep);
```

Syntaxe Basic interne

```
Function AmDbGetList(strQuery As String, strColSep As String,
strLineSep As String, strIdSep As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une liste, le résultat de l'exécution d'une requête AQL. Le nombre d'éléments sélectionnés par la requête AQL est limité à 99.

Entrée

- *strQuery* : Ce paramètre contient la requête AQL que vous souhaitez exécuter.
- *strColSep* : Ce paramètre contient le caractère utilisé comme séparateur de colonnes dans le résultat donné par la fonction.
- *strLineSep* : Ce paramètre contient le caractère utilisé comme séparateur de lignes dans le résultat donné par la fonction.
- *strIdSep* : Ce paramètre contient le caractère utilisé comme séparateur d'identifiant dans le résultat donné par la fonction.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetListEx()

Syntaxe API

```
long AmDbGetListEx(long hApiCnxBase, char *strQuery, char *pstrResult,
long lResult, char *strColSep, char *strLineSep, char *strIdSep);
```

Syntaxe Basic interne

```
Function AmDbGetListEx(strQuery As String, strColSep As String,
strLineSep As String, strIdSep As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une liste, le résultat de l'exécution d'une requête AQL. A la différence de la fonction `AmDbGetList`, cette fonction n'est pas limitée dans le nombre d'éléments sélectionnés par la requête AQL.

Entrée

- *`strQuery`* : Ce paramètre contient la requête AQL que vous souhaitez exécuter.
- *`strColSep`* : Ce paramètre contient le caractère utilisé comme séparateur de colonnes dans le résultat donné par la fonction.

- *strLineSep* : Ce paramètre contient le caractère utilisé comme séparateur de lignes dans le résultat donné par la fonction.
- *strIdSep* : Ce paramètre contient le caractère utilisé comme séparateur d'identifiant dans le résultat donné par la fonction.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetLong()

Syntaxe API

```
long AmDbGetLong(long hApiCnxBase, char *strQuery);
```

Syntaxe Basic interne

```
Function AmDbGetLong(strQuery As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le résultat de l'exécution d'une requête AQL.

Entrée

- *strQuery* : Ce paramètre contient l'intégralité de la requête AQL dont on veut récupérer le résultat.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant renvoie le numéro d'identifiant d'un fournisseur de produit :

```
AmDbGetLong("SELECT lSuppId FROM amProdSupp WHERE
lProdId="+Str([ProdId])+")
```

AmDbGetPk()

Syntaxe API

```
long AmDbGetPk(long hApiCnxBase, char *strTableName, char *strWhere);
```

Syntaxe Basic interne

```
Function AmDbGetPk(strTableName As String, strWhere As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la clé primaire d'une table en fonction de la clause WHERE d'une requête AQL.

Entrée

- *strTableName* : Nom SQL de la table dont on veut récupérer la clé primaire.
- *strWhere* : Clause WHERE d'une requête AQL.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetString()

Syntaxe API

```
long AmDbGetString(long hApiCnxBase, char *strQuery, char *pstrResult,
long lResult, char *strColSep, char *strLineSep);
```

Syntaxe Basic interne

```
Function AmDbGetString(strQuery As String, strColSep As String,
strLineSep As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne formatée, le résultat de l'exécution d'une requête AQL. Le nombre d'éléments sélectionnés par la requête AQL est limité à 99.

Entrée

- `strQuery` : Ce paramètre contient la requête AQL que vous souhaitez exécuter.
- `strColSep` : Ce paramètre contient le caractère utilisé comme séparateur de colonnes dans la chaîne finale.
- `strLineSep` : Ce paramètre contient le caractère utilisé comme séparateur de lignes dans la chaîne finale.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDbGetStringEx()

Syntaxe API

```
long AmDbGetStringEx(long hApiCnxBase, char *strQuery, char
*pstrResult, long lResult, char *strColSep, char *strLineSep);
```

Syntaxe Basic interne

```
Function AmDbGetStringEx(strQuery As String, strColSep As String,
strLineSep As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne formatée, le résultat de l'exécution d'une requête AQL. A la différence de la fonction

AmDbGetString, cette fonction n'est pas limitée dans le nombre d'éléments sélectionnés par la requête AQL.

Entrée

- *strQuery* : Ce paramètre contient la requête AQL que vous souhaitez exécuter.
- *strColSep* : Ce paramètre contient le caractère utilisé comme séparateur de colonnes dans la chaîne finale.
- *strLineSep* : Ce paramètre contient le caractère utilisé comme séparateur de lignes dans la chaîne finale.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction AmLastError (et éventuellement la fonction AmLastErrorMsg) pour savoir si une erreur s'est produite (et son message associé).

AmDeadLine()

Syntaxe API

```
long AmDeadLine(char *strCalendarSqlName, long tmStart, long
tsDuration);
```

Syntaxe Basic interne

```
Function AmDeadLine(strCalendarSqlName As String, tmStart As Date,
tsDuration As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction calcule une date d'échéance en fonction d'un calendrier, d'une date de début et d'un nombre de secondes ouvrées écoulées.

Entrée

- `strCalendarSqlName` : Ce paramètre contient le nom SQL du calendrier des périodes ouvrées utilisé comme base pour le calcul de la date d'échéance.
- `tmStart` : Ce paramètre contient la date de début de la période.
- `tsDuration` : Ce paramètre contient le nombre de secondes ouvrées écoulées à partir de la date de début de période.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant calcule la date d'échéance en fonction du calendrier de nom SQL "Calendar_Paris", d'une date de début de période fixée au 01/09/1998 à 8h00 et d'un nombre de secondes écoulées de 450000.

```
AmDeadLine("Calendar_Paris", "1998/09/01 08:00:00", 450000)
```

Cet exemple renvoie la date d'échéance, à savoir le 22/09/1998 à 18h00.

AmDecrementLogLevel()

Syntaxe API

```
long AmDecrementLogLevel();
```

Syntaxe Basic interne

```
Function AmDecrementLogLevel() As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de	

	Utilisable
caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction permet de remonter d'un cran dans l'arborescence d'une fenêtre d'historique.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDefAssignee()

Syntaxe API

```
long AmDefAssignee(long hApiCnxBase, long lGroupId);
```

Syntaxe Basic interne

```
Function AmDefAssignee(lGroupId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction recherche le numéro d'identifiant du chargé de groupe par défaut pour un groupe de support donné.

Entrée

- `lGroupId` : Ce paramètre contient le numéro d'identifiant d'un groupe de support.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple générique suivant renvoie l'identifiant du chargé par défaut d'un groupe de support :

```
AmDefAssignee([lGroupId])
```

Vous pouvez directement saisir la valeur numérique de l'identifiant, comme dans cet exemple :

```
AmDefAssignee(24)
```

AmDefaultCurrency()

Syntaxe Basic interne

```
Function AmDefaultCurrency() As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction renvoie la devise par défaut utilisée sous AssetCenter.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmDefEscalationScheme()

Syntaxe API

```
long AmDefEscalationScheme(long hApiCnxBase, char *strLocFullName, long
lSeverityLvl);
```

Syntaxe Basic interne

```
Function AmDefEscalationScheme(strLocFullName As String, lSeverityLvl
As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction recherche la procédure d'escalade par défaut en fonction de la localisation et de la gravité du dossier de support.

Entrée

- *`strLocFullName`* : Ce paramètre contient le nom complet de la localisation.
- *`lSeverityLvl`* : Ce paramètre contient la valeur de la gravité.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple générique suivant renvoie l'identifiant de la procédure d'escalade par défaut en fonction de la localisation et de la gravité :

```
AmDefEscalationScheme([Asset.Location.FullName],
[Severity.lSeverityLvl])
```

Vous pouvez directement saisir les valeurs des paramètres, comme dans cet exemple :

```
AmDefEscalationScheme ( "/Siège/", 24)
```

AmDefGroup()

Syntaxe API

```
long AmDefGroup(long hApiCnxBase, long lProblemClassId, char
*strLocFullName, long lAssetMainCntId);
```

Syntaxe Basic interne

```
Function AmDefGroup(lProblemClassId As Long, strLocFullName As String,
lAssetMainCntId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le numéro d'identifiant du groupe de support par défaut en fonction d'un type de problème, d'une localisation et d'un contrat de maintenance.

Entrée

- *lProblemClassId* : Ce paramètre contient le numéro d'identifiant d'un type de problème.
- *strLocFullName* : Ce paramètre contient le nom complet d'une localisation.
- *lAssetMainCntId* : Ce paramètre contient le numéro d'identifiant d'un contrat de maintenance.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple générique suivant calcule le numéro d'identifiant du groupe de support par défaut en fonction des trois paramètres : type de problème, localisation et contrat de maintenance.

```
AmDefGroup([ProblemClass.lPbClassId],[Asset.Location.FullName],
[Asset.lMaintCntrId])
```

Vous pouvez directement saisir la valeur numérique des paramètres utilisant des numéros d'identifiant, comme dans cet exemple :

```
AmDefGroup(0, [Asset.Location.FullName], 0)
```

AmDeleteLink()

Syntaxe API

```
long AmDeleteLink(long hApiRecord, char *strLinkName, long
hApiRecDest);
```

Syntaxe Basic interne

```
Function AmDeleteLink(hApiRecord As Long, strLinkName As String,
hApiRecDest As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction détruit un lien d'un enregistrement.

Entrée

Ce paramètre contient le handle sur l'enregistrement contenant le lien à détruire.Ce paramètre contient le nom SQL du lien à détruire.Ce paramètre contient un handle sur l'enregistrement de destination du lien qui doit être détruit.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmDeleteRecord()

Syntaxe API

```
long AmDeleteRecord(long hApiRecord);
```

Syntaxe Basic interne

```
Function AmDeleteRecord(hApiRecord As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X

	Utilisable
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction détruit un enregistrement dans la base de données.

Entrée

- *hApiRecord* : Ce paramètre contient un handle sur l'enregistrement que vous souhaitez détruire.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmEndOfNthBusinessDay()

Syntaxe API

```
long AmEndOfNthBusinessDay(char *strCalendarSqlName, long tmStart, long
lDayCount);
```

Syntaxe Basic interne

```
Function AmEndOfNthBusinessDay(strCalendarSqlName As String, tmStart As
Date, lDayCount As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Donne la dernière heure ouvrée du nième jour (identifié par l'entier `lDayCount`) à compter d'une date donnée et en respectant un calendrier.

Entrée

- `strCalendarSqlName` : Nom du calendrier utilisé pour le calcul.
- `tmStart` : Date de début du calcul.
- `lDayCount` : Nombre de jours ouvrés entiers à ajouter à dStart pour le calcul.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmEnumValList()

Syntaxe API

```
long AmEnumValList(long hApiCnxBase, char *strTableSqlName, char
*strEnumName, char *pstrValList, long lValList, long bNoCase, char
*strLineSep);
```

Syntaxe Basic interne

```
Function AmEnumValList(strTableSqlName As String, strEnumName As
String, bNoCase As Long, strLineSep As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie une chaîne contenant toutes les valeurs d'une énumération système. Les différentes valeurs sont délimitées par le séparateur indiqué dans le paramètre *strLineSep*.

Dans le cas où une valeur de l'énumération contient le caractère utilisé comme séparateur ou un "\", le préfixe "\" est utilisé.

Entrée

- *strTableSqlName* : Ce paramètre contient le nom SQL de la table contenant l'énumération système.
- *strEnumName* : Ce paramètre contient le nom SQL de l'énumération système dont vous souhaitez récupérer les valeurs.
- *bNoCase* : Ce paramètre ????
- *strLineSep* : Ce paramètre contient le caractère utilisé pour délimiter les valeurs de l'énumération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmExecTransition()

Syntaxe API

```
long AmExecTransition(char *strTransName);
```

Syntaxe Basic interne

```
Function AmExecTransition(strTransName As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction déclenche une transition valide à partir de la page en cours.

Entrée

- `strTransName` : Ce paramètre contient le nom de la transition tel qu'il est défini dans le script de l'assistant.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmExecuteActionById()

Syntaxe API

```
long AmExecuteActionById(long lActionId, char *strTableName, long
lRecordId);
```

Syntaxe Basic interne

```
Function AmExecuteActionById(lActionId As Long, strTableName As String,
lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction exécute une action identifiée par son identifiant.

Entrée

- *lActionId* : Ce paramètre contient l'identifiant de l'action à exécuter.
- *strTableName* : Dans le cas d'une action contextuelle, ce paramètre contient le nom SQL de la table sur laquelle l'action est exécutée. Si ce paramètre est omis, dans le cas d'une action contextuelle, la fonction échouera. Pour une action non contextuelle, ce paramètre n'est pas interprété et donc facultatif.
- *lRecordId* : Ce paramètre contient l'identifiant de l'enregistrement sur lequel porte éventuellement l'action. Pour une action non contextuelle, ce paramètre n'est pas interprété et donc facultatif.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmExecuteActionByName()

Syntaxe API

```
long AmExecuteActionByName(char *strSqlName, char *strTableName, long
lRecordId);
```

Syntaxe Basic interne

```
Function AmExecuteActionByName(strSqlName As String, strTableName As
String, lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction exécute une action identifiée par son nom SQL.

Entrée

- `strSqlName` : Ce paramètre contient le nom SQL de l'action à exécuter.
- `strTableName` : Dans le cas d'une action contextuelle, ce paramètre contient le nom SQL de la table sur laquelle l'action est exécutée. Si ce paramètre est omis, dans le cas d'une action contextuelle, la fonction échouera. Pour une action non contextuelle, ce paramètre n'est pas interprété et donc facultatif.
- `lRecordId` : Ce paramètre contient l'identifiant de l'enregistrement sur lequel porte éventuellement l'action. Pour une action non contextuelle, ce paramètre n'est pas interprété et donc facultatif.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmGenSqlName()

Syntaxe API

```
long AmGenSqlName(char *return, long lreturn, char *strText);
```

Syntaxe Basic interne

```
Function AmGenSqlName(strText As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction génère un nom SQL valide à partir d'une chaîne texte classique. Les espaces sont remplacés par des underscores ("_"). Cette fonction est particulièrement utile pour définir la valeur par défaut du nom SQL d'une caractéristique à partir de son nom.

Entrée

- *strText* : Chaîne de caractères à partir de laquelle vous souhaitez générer un nom SQL.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant définit la valeur par défaut du nom SQL d'un objet de nom "Label" de la base de données AssetCenter :

```
RetVal=AmSQLName([Label])
```

AmGetComputeString()

Syntaxe API

```
long AmGetComputeString(long hApiCnxBase, char *strTableName, long
lRecordId, char *strTemplate, char *pstrComputeString, long
lComputeString);
```

Syntaxe Basic interne

```
Function AmGetComputeString(strTableName As String, lRecordId As Long,
strTemplate As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la chaîne de description d'un enregistrement donné en fonction d'un modèle.

Entrée

- `strTableName` : Ce paramètre contient le nom SQL de la table de l'enregistrement dont on souhaite récupérer la chaîne de description.
- `lRecordId` : Ce paramètre contient l'identifiant de l'enregistrement au sein de la table.
- `strTemplate` : Ce paramètre contient, sous forme de chaîne de caractères, le modèle utilisé pour la chaîne de description.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFeat()

Syntaxe Basic interne

```
Function AmGetFeat(hApiTable As Long, lPos As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction créée un objet caractéristique à partir du handle d'une table et retourne le handle de l'objet caractéristique créé.

Entrée

- `hApiTable` : Ce paramètre contient un handle sur une table.
- `lPos` : Ce paramètre contient la position de la caractéristique à l'intérieur de la table.

AmGetFeatCount()

Syntaxe Basic interne

```
Function AmGetFeatCount(hApiTable As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction renvoie le nombre de caractéristiques sur la table précisée dans le paramètre `hApiTable`.

Entrée

- *hApiTable* : Ce paramètre contient un handle sur une table.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetField()

Syntaxe API

```
long AmGetField(long hApiObject, long lPos);
```

Syntaxe Basic interne

```
Function AmGetField(hApiObject As Long, lPos As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de	X

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet champ à partir du handle d'une requête, d'un enregistrement ou d'une table et retourne le handle de l'objet champ créé.

Entrée

- `hApiObject` : Ce paramètre contient un handle sur une requête, un enregistrement ou une table.
- `lPos` : Ce paramètre contient la position du champ (son index) à l'intérieur de l'objet.

AmGetFieldBlobLength()

Syntaxe API

```
long AmGetFieldBlobLength(long hApiRecord, long lFieldId);
```

Syntaxe Basic interne

```
Function AmGetFieldBlobLength(hApiRecord As Long, lFieldId As Long) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Récupère la taille d'un champ de la table système (Blob). Un "blob" est une valeur binaire de la base de données, par exemple une image.

Entrée

- `hApiRecord` : Handle valide sur un enregistrement.
- `lFieldId` : Identifiant du champ concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldBlobValue()

Syntaxe API

```
long AmGetFieldBlobValue(long hApiRecord);
```

Syntaxe Basic interne

```
Function AmGetFieldBlobValue(hApiRecord As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la valeur d'un champ blob. Cette fonction ne peut être appelée qu'après `AmGetFieldBlobLength`.

Entrée

- *hApiRecord* : Handle valide sur un enregistrement.

AmGetFieldCount()

Syntaxe API

```
long AmGetFieldCount(long hApiObject);
```

Syntaxe Basic interne

```
Function AmGetFieldCount(hApiObject As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nombre de champs contenus dans l'objet courant.

Entrée

- *hApiObject* : Ce paramètre contient un handle sur un enregistrement, une requête ou une table valides.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldDateValue()

Syntaxe API

```
long AmGetFieldDateValue(long hApiObject, long lFieldPos);
```

Syntaxe Basic interne

```
Function AmGetFieldDateValue(hApiObject As Long, lFieldPos As Long) As
Date
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la valeur d'un champ contenu dans l'objet courant. Cette valeur est renvoyée au format "Date".

Entrée

- `hApiObject` : Ce paramètre contient un handle sur une requête ou un enregistrement.
- `lFieldPos` : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldDescription()

Syntaxe API

```
long AmGetFieldDescription(long hApiField, char *pstrBuffer, long
lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldDescription(hApiField As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), la description d'un champ identifié par un handle.

Entrée

- *hApiField* : Ce paramètre contient un handle valide sur le champ dont on souhaite connaître la description longue.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldDoubleValue()

Syntaxe API

```
double AmGetFieldDoubleValue(long hApiObject, long lFieldPos);
```

Syntaxe Basic interne

```
Function AmGetFieldDoubleValue(hApiObject As Long, lFieldPos As Long)
As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la valeur d'un champ contenu dans l'objet courant. Cette valeur est renvoyée au format "Double".

Entrée

- *`hApiObject`* : Ce paramètre contient un handle sur une requête ou un enregistrement.
- *`lFieldPos`* : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldFormat()

Syntaxe API

```
long AmGetFieldFormat(long hApiField, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldFormat(hApiField As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction est utile quand le "UserType" (cf. fichier "database.txt") du champ concerné a pour valeur

- System itemized list

- Custom itemized list
- Time span
- Table or field SQL name

La fonction renvoie alors le format du "UserType", à savoir

UserType	Format renvoyé par la fonction
System Itemized List	Liste des entrées de l'énumération système.
Custom Itemized List	Nom de l'énumération associée au champ.
Time span	Format d'affichage.
Table or Field SQL Name	Nom SQL du champ qui stocke le nom SQL de la table contenant le champ que précise le champ décrit.

Entrée

- *hApiField* : Ce paramètre contient un handle valide sur le champ dont on souhaite connaître le "UserType".

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldFormatFromName()

Syntaxe API

```
long AmGetFieldFormatFromName(long hApiCnxBase, char *strTableName,
char *strFieldName, char *pFieldFormat, long lpFieldFormat);
```

Syntaxe Basic interne

```
Function AmGetFieldFormatFromName(strTableName As String, strFieldName
As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le format du "UserType" d'un champ, à partir de son nom.

Entrée

- *strTableName* : Ce paramètre contient le nom SQL de la table contenant le champ concerné par l'opération.
- *strFieldName* : Ce paramètre contient le nom SQL du champ.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldFromName()

Syntaxe API

```
long AmGetFieldFromName(long hApiObject, char *strName);
```

Syntaxe Basic interne

```
Function AmGetFieldFromName(hApiObject As Long, strName As String) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou	X

	Utilisable
d'un lien	
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet champ à partir de son nom et retourne le handle de l'objet champ créé.

Entrée

- *hApiObject* : Ce paramètre contient un handle sur une requête, un enregistrement ou une table.
- *strName* : Ce paramètre contient le nom du champ.

AmGetFieldLabel()

Syntaxe API

```
long AmGetFieldLabel(long hApiField, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldLabel(hApiField As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), le libellé d'un champ identifié par un handle.

Entrée

- `hApiField` : Ce paramètre contient un handle valide sur le champ dont on souhaite connaître le libellé.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldLabelFromName()

Syntaxe API

```
long AmGetFieldLabelFromName(long hApiCnxBase, char *strTableName, char
*strFieldName, char *pFieldLabel, long lpFieldLabel);
```

Syntaxe Basic interne

```
Function AmGetFieldLabelFromName(strTableName As String, strFieldName
As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le label d'un champ à partir de son nom SQL.

Entrée

- *strTableName* : Ce paramètre contient le nom SQL de la table contenant le champ concerné par l'opération.
- *strFieldName* : Ce paramètre contient le nom SQL du champ.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldLongValue()

Syntaxe API

```
long AmGetFieldLongValue(long hApiObject, long lFieldPos);
```

Syntaxe Basic interne

```
Function AmGetFieldLongValue(hApiObject As Long, lFieldPos As Long) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou	X

	Utilisable
d'un lien	
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la valeur d'un champ contenu dans l'objet courant.

Entrée

- *`hApiObject`* : Ce paramètre contient un handle sur une requête ou un enregistrement.
- *`lFieldPos`* : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Si vous utilisez cette fonction pour récupérer la valeur d'un champ de type date, heure ou date+heure, l'entier long renvoyé par la fonction

représente le nombre de secondes écoulées depuis le 01/01/1970 à 00:00:00.

AmGetFieldName()

Syntaxe API

```
long AmGetFieldName(long hApiObject, long lFieldPos, char *pstrBuffer,
long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldName(hApiObject As Long, lFieldPos As Long) As
String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nom d'un champ contenu dans l'objet courant.

Entrée

- *hApiObject* : Ce paramètre contient un handle sur une requête, un enregistrement ou une table.
- *lFieldPos* : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant. Par exemple, la valeur "0" désigne le premier champ.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldRights()

Syntaxe API

```
long AmGetFieldRights(long hApiObject, long lFieldPos, char
*pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldRights(hApiObject As Long, lFieldPos As Long) As
String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie les droits utilisateurs d'un champ de l'objet courant. Ces droits sont renvoyés sous la forme d'une chaîne composée de trois caractères précisant les droits en lecture/ insertion/ mise à jour :

- "r" : désigne l'autorisation en lecture.
- "i" : désigne l'autorisation en insertion.
- "u" : désigne l'autorisation en mise à jour.

Par exemple, pour un champ en lecture seule, la fonction renverra la valeur "r ".

Entrée

- `hApiObject` : Ce paramètre contient un handle sur une requête, un enregistrement ou une table.
- `lFieldPos` : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldSize()

Syntaxe API

```
long AmGetFieldSize(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetFieldSize(hApiField As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la taille d'un champ.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le champ dont on veut connaître la taille.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldSqlName()

Syntaxe API

```
long AmGetFieldSqlName(long hApiField, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldSqlName(hApiField As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), le nom SQL d'un champ identifié par un handle.

Entrée

- *`hApiField`* : Ce paramètre contient un handle valide sur le champ dont on souhaite connaître le nom SQL.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldStrValue()

Syntaxe API

```
long AmGetFieldStrValue(long hApiObject, long lFieldPos, char
*pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetFieldStrValue(hApiObject As Long, lFieldPos As Long) As
String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la valeur d'un champ contenu dans l'objet courant. Cette valeur est renvoyée au format chaîne.

Attention : Quand cette fonction est utilisée au travers des APIs AssetCenter, elle attend deux paramètres supplémentaires *strBuffer* et *lBuffer*, qui définissent respectivement une chaîne de caractères utilisée comme buffer pour le stockage de la chaîne récupérée et la taille de ce buffer. La chaîne *strBuffer* doit être formatée (remplie de caractères) et posséder la taille définie par *lBuffer*. La portion de code suivante est incorrecte, la chaîne utilisée comme buffer n'étant pas dimensionnée :

```
Dim strBuffer as String

Dim lRec as Long

Dim lBuffer as Long

lBuffer=20

lRec=AmGetFieldStrValue(1, 0, strBuffer, lBuffer)
```

Voici la portion de code corrigée :

```
Dim strBuffer as String

Dim lRec as Long

Dim lBuffer as Long

strBuffer=String(21, " ") ' Le buffer est dimensionné à 21
caractères (" ")

lBuffer=20

lRec=AmGetFieldStrValue(1, 0, strBuffer, lBuffer)
```

Lorsque vous formatez la chaîne du buffer au moyen de la fonction "String", n'utilisez jamais "0" comme caractère de remplissage. Dimensionnez le buffer avant chaque appel de la fonction `AmGetFieldStrValue`, en particulier si cette fonction se trouve dans une boucle et utilise toujours la même chaîne comme buffer.

Entrée

- *hApiObject* : Ce paramètre contient un handle sur une requête ou un enregistrement.
- *lFieldPos* : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetFieldType()

Syntaxe API

```
long AmGetFieldType(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetFieldType(hApiField As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le type d'un champ.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le champ dont on veut connaître le type.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Le tableau ci-dessous donne la correspondance entre les valeurs retournées par la fonction `AmGetFieldType` et les types de champs :

Valeurs retournées	Type de champ correspondant
0	Non défini
1	Byte
2	Short
3	Long
4	Float
5	Double
6	String
7	Time stamp

Valeurs retournées	Type de champ correspondant
8	Bin
9	Blob
10	Date
11	Time
12	Memo

AmGetFieldUserType()

Syntaxe API

```
long AmGetFieldUserType(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetFieldUserType(hApiField As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X

	Utilisable
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le "UserType" (cf. fichier `database.txt`) d'un champ identifié par un handle, sous la forme d'un entier long. Les valeurs valides renvoyées sont récapitulées dans le tableau ci-dessous :

Valeur stockée	Valeur en clair
0	Default
1	Number
2	Yes/ No
3	Money
4	Date
5	Date+Time
7	System itemized list
8	Custom itemized list
10	Percentage
11	Time span
12	Table or field SQL name

Entrée

- *`hApiField`* : Ce paramètre contient un handle valide sur le champ dont on souhaite connaître le "UserType".

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetForeignKey()

Syntaxe API

```
long AmGetForeignKey(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetForeignKey(hApiField As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Récupère le handle sur la clé externe d'un lien, lui même identifié par son handle.

Entrée

- `hApiField` : Handle sur le lien concerné par l'opération.

AmGetIndex()

Syntaxe Basic interne

```
Function AmGetIndex(hApiTable As Long, lPos As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?

	Utilisable
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction créée un objet index à partir du handle d'une requête, d'un enregistrement ou d'une table et retourne le handle de l'objet index créé.

Entrée

- `hApiTable` : Ce paramètre contient un handle sur une table.
- `lPos` : Ce paramètre contient la position de l'index à l'intérieur de la table.

AmGetIndexCount()

Syntaxe Basic interne

```
Function AmGetIndexCount(hApiTable As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction renvoie le nombre d'index contenus dans la table précisée dans le paramètre *hApiTable*.

Entrée

- *hApiTable* : Ce paramètre contient un handle sur une table.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetIndexField()

Syntaxe Basic interne

```
Function AmGetIndexField(hApiIndex As Long, lPos As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

AmGetIndexFieldCount()

Syntaxe Basic interne

```
Function AmGetIndexFieldCount(hApiIndex As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetIndexFlags()

Syntaxe Basic interne

```
Function AmGetIndexFlags(hApiIndex As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetIndexName()

Syntaxe Basic interne

```
Function AmGetIndexName(hApiIndex As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	?

Description

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetLink()

Syntaxe API

```
long AmGetLink(long hApiTable, long lPos);
```

Syntaxe Basic interne

```
Function AmGetLink(hApiTable As Long, lPos As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet lien à partir du handle d'une table et retourne le handle de l'objet lien créé.

Entrée

- `hApiTable` : Ce paramètre contient un handle sur une table.
- `lPos` : Ce paramètre contient la position du lien (son index) à l'intérieur de l'objet.

AmGetLinkCardinality()

Syntaxe Basic interne

```
Function AmGetLinkCardinality(hApiField As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction renvoie la cardinalité d'un lien.

Entrée

- `hApiField` : Ce paramètre contient un handle sur le lien dont vous souhaitez connaître la cardinalité.

Sortie

- 1 : Le lien est de cardinalité 1-1.
- 2 : Le lien est de cardinalité 1-n.

AmGetLinkCount()

Syntaxe API

```
long AmGetLinkCount(long hApiTable);
```

Syntaxe Basic interne

```
Function AmGetLinkCount(hApiTable As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nombre de liens contenus dans la table courante.

Entrée

- *hApiTable* : Ce paramètre contient un handle sur une table valide.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetLinkDstField()

Syntaxe Basic interne

```
Function AmGetLinkDstField(hApiField As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction renvoie le champ sur lequel pointe le lien définit par le paramètre *hApiField*.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le lien concerné par l'opération.

AmGetLinkFeatureValue()

Syntaxe API

```
long AmGetLinkFeatureValue(long hApiObject, long lFieldPos, long
lRecordId);
```

Syntaxe Basic interne

```
Function AmGetLinkFeatureValue(hApiObject As Long, lFieldPos As Long,
lRecordId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la valeur d'une caractéristique de type "Lien".

Entrée

- `hApiObject` : Ce paramètre contient un handle sur une requête ou un enregistrement.
- `lFieldPos` : Ce paramètre contient le numéro du champ à l'intérieur de l'objet courant.
- `lRecordId` : Ce paramètre contient le numéro d'identifitant de l'enregistrement dont on veut récupérer la valeur pour la caractéristique.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim q as String
q = "Select fv_link, lEmplDeptId From amEmplDept Where lEmplDeptId
= " & [lEmplDeptId]
Dim hq as Long
hq = amQueryCreate()
Dim lErr as Long
lErr = amQueryGet(hq, q)
Dim lId as Long
lId = amGetFieldLongValue(hq, 1)
print "str: " & amGetFieldStrValue(hq, 0)
print "int: " &
amGetFieldLongValue(hq,0)
print "lnk: " & amGetLinkFeatureValue(hq,0,lId)
```

AmGetLinkFromName()

Syntaxe API

```
long AmGetLinkFromName(long hApiTable, char *strName);
```

Syntaxe Basic interne

```
Function AmGetLinkFromName(hApiTable As Long, strName As String) As
Long
```

Champ d'application

Version : 3.02

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet lien à partir de son nom et retourne le handle de l'objet lien créé.

Entrée

- `hApiTable` : Ce paramètre contient un handle sur une table.
- `strName` : Ce paramètre contient le nom SQL du lien.

AmGetLinkType()

Syntaxe API

```
long AmGetLinkType(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetLinkType(hApiField As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le type d'un lien.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le lien dont on veut connaître le type.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetMainField()

Syntaxe API

```
long AmGetMainField(long hApiTable);
```

Syntaxe Basic interne

```
Function AmGetMainField(hApiTable As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet champ, correspondant au champ principal d'une table donnée. Elle renvoie un handle sur le champ ainsi créé.

Entrée

- *hApiTable* : Ce paramètre contient un handle sur la table dont on cherche le champ principal.

AmGetRecordFromMainId()

Syntaxe API

```
long AmGetRecordFromMainId(long hApiCnxBase, char *strTable, long lId);
```

Syntaxe Basic interne

```
Function AmGetRecordFromMainId(strTable As String, lId As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction renvoie le numéro d'identifiant d'un enregistrement identifié par une valeur de la clé primaire de la table contenant cet enregistrement.

Entrée

- `strTable` : Ce paramètre contient le nom SQL de la table contenant l'enregistrement concerné.
- `lId` : Ce paramètre contient la valeur de la clé primaire de la table pour cet enregistrement.

AmGetRecordHandle()

Syntaxe API

```
long AmGetRecordHandle(long hApiQuery);
```

Syntaxe Basic interne

```
Function AmGetRecordHandle(hApiQuery As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction renvoie le handle d'un enregistrement qui est le résultat courant d'une requête identifiée par son handle. Cet enregistrement pourra être utilisé pour écrire dans la base de données .

Entrée

- *hApiQuery* : Ce paramètre contient un handle valide sur un objet requête.

AmGetRecordId()

Syntaxe API

```
long AmGetRecordId(long hApiRecord);
```

Syntaxe Basic interne

```
Function AmGetRecordId(hApiRecord As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction renvoie le numéro d'identifiant d'un enregistrement identifié par son handle.

Entrée

- `hApiRecord` : Ce paramètre contient un handle valide sur l'enregistrement dont on souhaite connaître le numéro d'identifiant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetReverseLink()

Syntaxe API

```
long AmGetReverseLink(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetReverseLink(hApiField As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le handle du lien inverse du lien spécifié par le handle contenu dans le paramètre *hApiField*.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le lien dont on veut connaître le lien inverse.

AmGetSelfFromMainId()

Syntaxe API

```
long AmGetSelfFromMainId(long hApiCnxBase, char *strTableName, long
lId, char *pstrRecordDesc, long lRecordDesc);
```

Syntaxe Basic interne

```
Function AmGetSelfFromMainId(strTableName As String, lId As Long) As
String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de	X

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la chaîne de description pour un enregistrement d'une table donnée.

Entrée

- `strTableName` : Ce paramètre contient le nom SQL de la table contenant l'enregistrement concerné par l'opération.
- `lId` : Ce paramètre contient le numéro d'identifiant de l'enregistrement concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetSourceTable()

Syntaxe API

```
long AmGetSourceTable(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetSourceTable(hApiField As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le handle de la table source du lien indiqué dans le paramètre *hApiField*.

Entrée

- *hApiField* : Ce paramètre contient un handle valide sur le lien dont on veut connaître la table source.

Sortie

En cas d'erreur, cette fonction retourne un handle non valide (de valeur nulle).

AmGetTable()

Syntaxe API

```
long AmGetTable(long hApiCnxBase, long lPos);
```

Syntaxe Basic interne

```
Function AmGetTable(lPos As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le handle d'une table identifiée par sa position (son numéro) dans la connexion courante.

Entrée

- *lPos* : Ce paramètre contient la position de la table dans la connexion courante. Ses valeurs sont comprises entre "0" et AmGetTableCount.

Sortie

En cas d'erreur, cette fonction retourne un handle non valide (de valeur nulle).

AmGetTableCount()

Syntaxe API

```
long AmGetTableCount(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmGetTableCount() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nombre de tables de la base de données sur laquelle porte la connexion courante.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTableDescription()

Syntaxe API

```
long AmGetTableDescription(long hApiTable, char *pstrDesc, long lDesc);
```

Syntaxe Basic interne

```
Function AmGetTableDescription(hApiTable As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), la description longue d'une table identifiée par un handle.

Entrée

- *hApiTable* : Ce paramètre contient un handle valide sur la table dont on souhaite connaître la description longue.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTableFromName()

Syntaxe API

```
long AmGetTableFromName(long hApiCnxBase, char *strName);
```

Syntaxe Basic interne

```
Function AmGetTableFromName(strName As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le handle d'une table identifiée par son nom SQL dans la connexion courante.

Entrée

- *strName* : Ce paramètre contient la position le nom SQL de la table dont on veut récupérer le handle.

Sortie

En cas d'erreur, cette fonction retourne un handle non valide (de valeur nulle).

AmGetTableLabel()

Syntaxe API

```
long AmGetTableLabel(long hApiTable, char *pstrLabel, long lLabel);
```

Syntaxe Basic interne

```
Function AmGetTableLabel(hApiTable As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), le libellé d'une table identifiée par un handle.

Entrée

- `hApiTable` : Ce paramètre contient un handle valide sur la table dont on souhaite connaître le libellé.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTableName()

Syntaxe API

```
long AmGetTableName(long hApiTable, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetTableName(hApiTable As Long) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le nom SQL d'une table sous la forme d'une chaîne de caractères.

Entrée

- *hApiTable* : Handle valide sur la table dont vous souhaitez récupérer le nom.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTableRights()

Syntaxe API

```
long AmGetTableRights(long hApiTable, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetTableRights(hApiTable As Long) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), les droits utilisateur sur une table identifiée par un handle. La chaîne renvoyée est composée au maximum de deux caractères qui indiquent l'état des droits en création et en destruction :

- "c" indique que l'utilisateur possède les droits en création sur la table.
- "d" indique que l'utilisateur possède les droits en destruction sur la table.

Ainsi, par exemple :

- " c" indique que l'utilisateur possède uniquement un droit en création sur la table.
- "cd" indique que l'utilisateur possède les droits en création et en destruction sur la table.

Entrée

- `hApiTable` : Ce paramètre contient un handle valide sur la table pour laquelle on souhaite connaître les droits utilisateurs.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTableSqlName()

Syntaxe API

```
long AmGetTableSqlName(long hApiTable, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmGetTableSqlName(hApiTable As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie, sous la forme d'une chaîne de caractères (format "String"), le nom SQL d'une table identifiée par un handle.

Entrée

- *hApiTable* : Ce paramètre contient un handle valide sur la table dont on souhaite connaître le nom SQL.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmGetTargetTable()

Syntaxe API

```
long AmGetTargetTable(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetTargetTable(hApiField As Long) As Long
```

Champ d'application

Version : ?

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Retourne le nom SQL de la table de destination d'un lien.

Entrée

- `hApiField` : Handle sur le lien concerné par l'opération.

Sortie

En cas d'erreur, cette fonction retourne un handle non valide (de valeur nulle).

AmGetTypedLinkField()

Syntaxe API

```
long AmGetTypedLinkField(long hApiField);
```

Syntaxe Basic interne

```
Function AmGetTypedLinkField(hApiField As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie un handle sur le champ dont la valeur est le nom SQL de la table de destination du lien typé indiqué dans le paramètre *hApiField*.

Entrée

- *hApiField* : Ce paramètre contient un handle valide sur le lien typé à l'origine de l'opération.

AmGetVersion()

Syntaxe API

```
long AmGetVersion(char *pstrBuf, long lBuf);
```

Syntaxe Basic interne

```
Function AmGetVersion() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le numéro de compilation de la version d'AssetCenter sous la forme d'une chaîne de caractères.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmHasAdminPrivilege()

Syntaxe API

```
long AmHasAdminPrivilege(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmHasAdminPrivilege() As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la valeur "TRUE" (=1) si l'utilisateur connecté possède les droits administratifs.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmIncrementLogLevel()

Syntaxe API

```
long AmIncrementLogLevel(char *strMsg, long iType);
```

Syntaxe Basic interne

```
Function AmIncrementLogLevel(strMsg As String, iType As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction affiche le message `strMsg` dans une fenêtre d'historique et créée un nœud.

Tous les prochains messages apparaîtront sous ce nœud.

Entrée

- `strMsg` : Ce paramètre contient le texte du message à afficher.
- `iType` : Ce paramètre définit l'icône associée au message. Les valeurs possibles sont "1" pour une erreur, "2" pour un avertissement et "4" pour une information.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmInsertRecord()

Syntaxe API

```
long AmInsertRecord(long hApiRecord);
```

Syntaxe Basic interne

```
Function AmInsertRecord(hApiRecord As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction insère un enregistrement précédemment créé dans la base de données. Seuls les enregistrements créés au moyen de la fonction `AmCreateRecord` peuvent être insérés dans la base de données. Les enregistrements accédés au moyen d'une requête ne peuvent être insérés.

Entrée

- *`hApiRecord`* : Ce paramètre contient un handle sur l'enregistrement que vous souhaitez insérer dans la base de données.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmIsConnected()

Syntaxe API

```
long AmIsConnected(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmIsConnected() As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X

	Utilisable
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction teste si la connexion courante est valide.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmIsFieldForeignKey()

Syntaxe API

```
long AmIsFieldForeignKey(long hApiField);
```

Syntaxe Basic interne

```
Function AmIsFieldForeignKey(hApiField As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction permet de déterminer si un champ est une clé étrangère de la base de données.

Entrée

- *`hApiField`* : Ce paramètre contient un handle sur le champ qui doit être identifié.

Sortie

- 1 : Le champ est une clé étrangère.
- 0 : Le champ n'est pas une clé étrangère.

AmIsFieldIndexed()

Syntaxe Basic interne

```
Function AmIsFieldIndexed(hApiField As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	?
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	?
Script de définition d'un paramètre de caractéristique	?
Action de type "Script"	?
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	?

Description

Cette fonction permet de déterminer si un champ est indexé ou non.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le champ qui doit être identifié.

Sortie

- 1 : Le champ est indexé.
- 0 : Le champ n'est pas indexé.

AmIsFieldPrimaryKey()

Syntaxe API

```
long AmIsFieldPrimaryKey(long hApiField);
```

Syntaxe Basic interne

```
Function AmIsFieldPrimaryKey(hApiField As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction permet de déterminer si un champ est une clé primaire de la base de données.

Entrée

- *hApiField* : Ce paramètre contient un handle sur le champ qui doit être identifié.

Sortie

- 1 : Le champ est une clé primaire.
- 0 : Le champ n'est pas une clé primaire.

AmIsLink()

Syntaxe API

```
long AmIsLink(long hApiField);
```

Syntaxe Basic interne

```
Function AmIsLink(hApiField As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Détermine si l'objet identifié par son handle est un lien ou un champ.

Entrée

- `hApiField` : Handle sur l'objet concerné par l'opération.

Sortie

- 1 : L'objet est un lien.
- 0 : L'objet est un champ.

AmIsTypedLink()

Syntaxe API

```
long AmIsTypedLink(long hApiField);
```

Syntaxe Basic interne

```
Function AmIsTypedLink(hApiField As Long) As Long
```

Champ d'application

Version : 3.02

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Détermine si l'objet identifié par son handle est un lien typé ou non.

Entrée

- `hApiField` : Handle sur l'objet concerné par l'opération.

Sortie

- 1 : L'objet est un lien typé.
- 0 : L'objet n'est pas un lien typé.

AmLastError()

Syntaxe API

```
long AmLastError(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmLastError() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le dernier code d'erreur généré par la dernière fonction exécutée dans le contexte de la connexion correspondante.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmLastErrorMsg()

Syntaxe API

```
long AmLastErrorMsg(long hApiCnxBase, char *pstrBuffer, long lBuffer);
```

Syntaxe Basic interne

```
Function AmLastErrorMsg() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le dernier message d'erreur survenu lors de la connexion courante.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmListToString()

Syntaxe API

```
long AmListToString(char *return, long lreturn, char *strSource, char
*strColSep, char *strLineSep, char *strIdSep);
```

Syntaxe Basic interne

```
Function AmListToString(strSource As String, strColSep As String,
strLineSep As String, strIdSep As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit le résultat d'une chaîne de caractères obtenue au moyen de la fonction `AmDbGetList` en une chaîne de caractères affichable telle qu'elle apparaîtrait avec la fonction `AmDbGetString`.

Entrée

- *`strSource`* : Ce paramètre contient la chaîne de caractères à convertir.
- *`strColSep`* : Ce paramètre contient le caractère utilisé comme séparateur de colonnes dans la chaîne à convertir.
- *`strLineSep`* : Ce paramètre contient le caractère utilisé comme séparateur de lignes dans la chaîne à convertir.
- *`strIdSep`* : Ce paramètre contient le caractère utilisé comme séparateur d'identifiant dans la chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmLog()

Syntaxe API

```
long AmLog(char *strMessage, long iLogType);
```

Syntaxe Basic interne

```
Function AmLog(strMessage As String, iLogType As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction affiche le message *strMessage* dans une fenêtre d'historique.

Entrée

- *strMessage* : Ce paramètre contient le texte du message à afficher.
- *iLogType* : Ce paramètre définit l'icône associée au message. Les valeurs possibles sont "1" pour une erreur, "2" pour un avertissement et "4" pour une information.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
AmLog("Ceci est un message")
```

AmLoginId()

Syntaxe API

```
long AmLoginId(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmLoginId() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie l'identifiant de l'utilisateur connecté.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant définit l'identifiant de l'utilisateur connecté comme valeur par défaut pour un champ de la base de données :

```
RetVal=AmLoginId()
```

AmLoginName()

Syntaxe API

```
long AmLoginName(long hApiCnxBase, char *return, long lreturn);
```

Syntaxe Basic interne

```
Function AmLoginName() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nom de login de l'utilisateur connecté.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant définit le nom de login de l'utilisateur connecté comme valeur par défaut pour un champ de la base de données :

```
RetVal=AmLoginName()
```

AmMsgBox()

Syntaxe API

```
long AmMsgBox(char *strMessage);
```

Syntaxe Basic interne

```
Function AmMsgBox(strMessage As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	

	Utilisable
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction affiche une boîte de dialogue contenant un message.

Entrée

- *strMessage* : Ce paramètre contient le message affiché dans la boîte de dialogue.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
AmMsgBox("Déménagement effectué")
```

AmOpenConnection()

Syntaxe API

```
long AmOpenConnection(char *strDataSource, char *strUser, char
*strPwd);
```

Syntaxe Basic interne

```
Function AmOpenConnection(strDataSource As String, strUser As String,
strPwd As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Crée une connection sur une base de données AssetCenter. `strDataSource` doit être une source de données valide (les sources de données apparaissent dans la boîte de connexion d'AssetCenter).

Vous pouvez ouvrir plusieurs connexions sur une même base ou sur des bases de données différentes.

Entrée

- `strDataSource` : Nom de la source de données pour la connexion.
- `strUser` : Nom de l'utilisateur pour le connexion.
- `strPwd` : Mot de passe de l'utilisateur sur la base de données.

AmPagePath()

Syntaxe API

```
long AmPagePath(char *pstrPath, long lPath);
```

Syntaxe Basic interne

```
Function AmPagePath() As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction renvoie, sous la forme d'une chaîne, le chemin de l'assistant, c'est-à-dire la liste des pages parcourues sans tenir compte des retours en arrière.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmProgress()

Syntaxe API

```
long AmProgress(long iProgress);
```

Syntaxe Basic interne

```
Function AmProgress(iProgress As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction affiche une barre de progression représentant un pourcentage.

Entrée

- *`iProgress`* : Ce paramètre contient le pourcentage (entre 0 et 100) de complétion qui détermine la taille de la barre de progression.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
AmProgress(85)
```

Cette fonction affiche une barre de progression représentant un pourcentage de 85%.

AmQueryCreate()

Syntaxe API

```
long AmQueryCreate(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmQueryCreate() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction créée un objet requête dans la connexion courante. Cet objet peut ensuite être utilisé pour envoyer des commandes AQL au serveur de base de données.

AmQueryExec()

Syntaxe API

```
long AmQueryExec(long hApiQuery, char *strQueryCommand);
```

Syntaxe Basic interne

```
Function AmQueryExec(hApiQuery As Long, strQueryCommand As String) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction exécute une requête AQL. Elle renvoie le premier résultat de la requête. Le résultat suivant peut être obtenu au moyen de la fonction `AmQueryNext`.

Lorsque la requête transmise par cette fonction renvoie un champ de type "Memo" (enregistrement de la table de nom SQL amComment), la taille de ce dernier est tronquée à 255 caractères.

Entrée

- *`hApiQuery`* : Ce paramètre contient un handle valide sur l'objet requête auquel sont transmises les commandes AQL.

- *strQueryCommand* : Ce paramètre contient le corps de la requête AQL sous forme de chaîne.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmQueryGet()

Syntaxe API

```
long AmQueryGet(long hApiQuery, char *strQueryCommand);
```

Syntaxe Basic interne

```
Function AmQueryGet(hApiQuery As Long, strQueryCommand As String) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction exécute une requête AQL. Elle ne renvoie que le premier résultat de la requête.

Entrée

- *hApiQuery* : Ce paramètre contient un handle valide sur l'objet requête auquel sont transmises les commandes AQL.
- *strQueryCommand* : Ce paramètre contient le corps de la requête AQL sous forme de chaîne.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmQueryNext()

Syntaxe API

```
long AmQueryNext(long hApiQuery);
```

Syntaxe Basic interne

```
Function AmQueryNext(hApiQuery As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le résultat suivant d'une requête préalablement exécutée au moyen de la fonction `AmQueryExec`.

Entrée

- *`hApiQuery`* : Ce paramètre contient un handle valide sur l'objet requête auquel sont transmises les commandes AQL.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmQueryStartTable()

Syntaxe API

```
long AmQueryStartTable(long hApiQuery);
```

Syntaxe Basic interne

```
Function AmQueryStartTable(hApiQuery As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie un handle sur la table sur laquelle porte une requête identifiée par son handle.

Entrée

- *hApiQuery* : Ce paramètre contient un handle valide sur un objet requête.

Sortie

En cas d'erreur, cette fonction retourne un handle non valide (de valeur nulle).

AmQueryStop()

Syntaxe API

```
long AmQueryStop(long hApiQuery);
```

Syntaxe Basic interne

```
Function AmQueryStop(hApiQuery As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction interrompt l'exécution d'une requête identifiée par son handle. Cette requête doit avoir été préalablement lancée au moyen de la fonction `AmQueryExec`.

Entrée

- *`hApiQuery`* : Ce paramètre contient un handle valide sur un objet requête.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmReceiveAllPOLines()

Syntaxe API

```
long AmReceiveAllPOLines(long hApiCnxBase, long lPOrdId, long
lDelivId);
```

Syntaxe Basic interne

```
Function AmReceiveAllPOLines(lPOrdId As Long, lDelivId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la réception de tous les éléments sur une ligne de commande.

Note: Attention : les lignes de réception sont créées par un agent au moment du "commit" de la transaction. Vous ne pouvez pas y accéder avant.

Entrée

- `lPOrdId` : Ce paramètre contient l'identifiant de la ligne de commande contenant les éléments à réceptionner.
- `lDelivId` : Ce paramètre contient l'identifiant de la fiche de réception qui recevra tous les éléments présents sur la ligne de commande.

Sortie

- 0 : La fonction s'est exécutée normalement.

- Non nul : Code d'erreur.

AmReceivePOLine()

Syntaxe API

```
long AmReceivePOLine(long hApiCnxBase, long lPOrdLineId, long lDelivId,
long lQty);
```

Syntaxe Basic interne

```
Function AmReceivePOLine(lPOrdLineId As Long, lDelivId As Long, lQty As
Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction effectue la réception d'une certaine quantité d'éléments sur une ligne de commande et renvoie le numéro d'identifiant de la ligne de réception.

Note: Attention : les lignes de réception sont créées par un agent au moment du "commit" de la transaction. Vous ne pouvez pas y accéder avant.

Entrée

- *lPOrdLineId* : Ce paramètre contient l'identifiant de la ligne de commande contenant les éléments à réceptionner.
- *lDelivId* : Ce paramètre contient l'identifiant de la fiche de réception qui recevra une certaine quantité des éléments présents sur la ligne de commande.
- *lQty* : Ce paramètre contient la quantité d'éléments sur la ligne de commande à réceptionner dans la fiche de réception.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmRefreshAllCaches()

Syntaxe API

```
long AmRefreshAllCaches(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmRefreshAllCaches() As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction rafraîchit l'ensemble des caches utilisés sous AssetCenter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmRefreshProperty()

Syntaxe API

```
long AmRefreshProperty(char *strVarName);
```

Syntaxe Basic interne

```
Function AmRefreshProperty(strVarName As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Réévalue la valeur d'une propriété identifiée par le paramètre *strVarName*. Si cette propriété utilise un script, celui-ci est à nouveau exécuté.

L'arbre de dépendance est remis à jour, le cas échéant.

Entrée

- *strVarName* : Nom de la propriété (de l'assistant) que vous souhaitez réévaluer.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmRefuseRequest()

Syntaxe API

```
long AmRefuseRequest(long hApiCnxBase, long lReqId, char *strComment);
```

Syntaxe Basic interne

```
Function AmRefuseRequest(lReqId As Long, strComment As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmReleaseHandle()

Syntaxe API

```
long AmReleaseHandle(long hApiObject);
```

Syntaxe Basic interne

```
Function AmReleaseHandle(hApiObject As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction libère le handle et tous les sous-handles d'un objet.

Entrée

- *`hApiObject`* : Ce paramètre contient un handle sur l'objet concerné.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmRgbColor()

Syntaxe API

```
long AmRgbColor(char *strText);
```

Syntaxe Basic interne

```
Function AmRgbColor(strText As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction donne la valeur RGB de la couleur correspondant au paramètre *strText*.

Entrée

- *strText*: Ce paramètre contient le nom d'une couleur :
  - √ Blanc
  - √ Gris clair
  - √ Gris
  - √ Gris foncé
  - √ Noir
  - √ Rouge

√ Vert
√ Bleu
√ Jaune
√ Cyan
√ Mauve
√ Jaune foncé<
√ Vert foncé
√ Cyan foncé
√ Bleu foncé
√ Mauve foncé
√ Rouge foncé

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmRollback()

Syntaxe API

```
long AmRollback(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmRollback() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction annule toutes les modifications effectuées avant la déclaration de début de transaction (effectuée via la fonction `AmStartTransaction`).

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSeCreateDefVal()

Syntaxe API

```
long AmSeCreateDefVal(long hApiCnxBase, long iProductTypeNature);
```

Syntaxe Basic interne

```
Function AmSeCreateDefVal(iProductTypeNature As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction fixe la valeur par défaut du champ de nom SQL "seCreate" présent dans les tables de noms SQL :

- amProdCompo
- amReqLine
- amPOrdLine
- amEstimLine

Entrée

- `iProductTypeNature` : Ce paramètre contient un entier représentant l'élément de l'énumération système associé au champ de nom SQL "seCreate". Le tableau ci-dessous donne pour chaque élément de l'énumération système, l'entier qui lui est associé :

Elément de l'énumération	Entier associé à l'élément
Matériel standard	0
Ordinateur	1
Licence logicielle	2
Intervention	3
Contrat	4
Configuration type	5
Formation	6
Autre	7

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetFieldDateValue()

Syntaxe API

```
long AmSetFieldDateValue(long hApiRecord, char *strFieldName, long
tmValue);
```

Syntaxe Basic interne

```
Function AmSetFieldDateValue(hApiRecord As Long, strFieldName As
String, tmValue As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction modifie un champ d'un enregistrement. Aucune mise à jour de la base de données n'est effectuée par cette fonction.

Entrée

- `hApiRecord` : Ce paramètre contient le handle sur l'enregistrement contenant le champ à modifier.
- `strFieldName` : Ce paramètre contient le nom SQL du champ à modifier.
- `tmValue` : Ce paramètre contient la nouvelle valeur du champ au format "Date".

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetFieldDoubleValue()

Syntaxe API

```
long AmSetFieldDoubleValue(long hApiRecord, char *strFieldName, double
dValue);
```

Syntaxe Basic interne

```
Function AmSetFieldDoubleValue(hApiRecord As Long, strFieldName As
String, dValue As Double) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction modifie en mémoire un champ d'un enregistrement. Aucune mise à jour de la base de données n'est effectuée par cette fonction.

Entrée

- *hApiRecord* : Ce paramètre contient le handle sur l'enregistrement contenant le champ à modifier.
- *strFieldName* : Ce paramètre contient le nom SQL du champ à modifier.
- *dValue* : Ce paramètre contient la nouvelle valeur du champ au format "Double".

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetFieldLongValue()

Syntaxe API

```
long AmSetFieldLongValue(long hApiRecord, char *strFieldName, long
lValue);
```

Syntaxe Basic interne

```
Function AmSetFieldLongValue(hApiRecord As Long, strFieldName As
String, lValue As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction modifie un champ d'un enregistrement. Aucune mise à jour de la base de données n'est effectuée par cette fonction. Pour modifier la valeur d'un champ date, heure ou date+heure, vous devez donner comme nouvelle valeur le nombre de secondes écoulées depuis le 01/01/1970 à 00:00:00.

Entrée

- `hApiRecord` : Ce paramètre contient le handle sur l'enregistrement contenant le champ à modifier.
- `strFieldName` : Ce paramètre contient le nom SQL du champ à modifier. Vous pouvez également donner le nom SQL d'une caractéristique, d'un champ de type "Commentaire" ou encore d'un champ de script.
- `lValue` : Ce paramètre contient la nouvelle valeur du champ.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetFieldStrValue()

Syntaxe API

```
long AmSetFieldStrValue(long hApiRecord, char *strFieldName, char
*strValue);
```

Syntaxe Basic interne

```
Function AmSetFieldStrValue(hApiRecord As Long, strFieldName As String,
strValue As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction modifie un champ d'un enregistrement. Aucune mise à jour de la base de données n'est effectuée par cette fonction.

Entrée

- *hApiRecord* : Ce paramètre contient le handle sur l'enregistrement contenant le champ à modifier.
- *strFieldName* : Ce paramètre contient le nom SQL du champ à modifier.
- *strValue* : Ce paramètre contient la nouvelle valeur du champ au format "String" (chaîne de caractères).

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetLinkFeatureValue()

Syntaxe API

```
long AmSetLinkFeatureValue(long hApiRecord, char *strFeatSqlName, char
*strDstSelfValue, long lDstId);
```

Syntaxe Basic interne

```
Function AmSetLinkFeatureValue(hApiRecord As Long, strFeatSqlName As
String, strDstSelfValue As String, lDstId As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	

	Utilisable
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction fixe la valeur d'une caractéristique de type lien pour un enregistrement donné.

Entrée

- `hApiRecord` : Ce paramètre contient l'identifiant de l'enregistrement auquel est associée la caractéristique de type lien.
- `strFeatSqlName` : Ce paramètre contient le nom SQL de la caractéristique de type lien dont on souhaite fixer la valeur. Ce nom SQL est toujours préfixé par "fv_".
- `strDstSelfValue` : Ce paramètre contient la valeur de la caractéristique telle qu'elle sera affichée pour l'enregistrement. Il s'agit de la valeur "Self" de l'enregistrement d'identifiant `lDstId`. Si vous renseignez ce paramètre avec une valeur non valide ou non existante, l'intégrité de la base de données risque d'être corrompue.
- `lDstId` : Ce paramètre contient l'identifiant de l'enregistrement sur lequel pointe la caractéristique de type lien.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetProperty()

Syntaxe API

```
long AmSetProperty(char *strVarName);
```

Syntaxe Basic interne

```
Function AmSetProperty(strVarName As String, vValue As Variant) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction fixe la valeur d'une propriété identifiée par son nom. Elle met également à jour l'arbre de dépendances de cette propriété.

Entrée

- *strVarName* : Ce paramètre contient le nom de la propriété dont on souhaite fixer la valeur.
- *vValue* : Ce paramètre contient la nouvelle valeur pour la propriété.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmSetUseUserName()

Syntaxe API

```
long AmSetUseUserName(long hApiCnxBase, long bUseUserName);
```

Syntaxe Basic interne

```
Function AmSetUseUserName(bUseUserName As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	

	Utilisable
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmStartTransaction()

Syntaxe API

```
long AmStartTransaction(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmStartTransaction() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X

	Utilisable
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction démarre une nouvelle transaction avec la base de données associée à la connexion. La prochaine commande de "Commit" ou de "Rollback" validera ou annulera toutes les modifications apportées à la base de données.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmStartup()

Syntaxe Basic interne

```
Champ d'application
```

Version : 2.52

	Utilisable

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction doit être appelée avant tout autre fonction. Elle initialise l'appel aux librairies AssetCenter.

AmTableDesc()

Syntaxe API

```
long AmTableDesc(long hApiCnxBase, char *return, long lreturn, char
*strSqlName);
```

Syntaxe Basic interne

```
Function AmTableDesc(strSqlName As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction génère une chaîne de format "<Description de la table> (<Nom SQL de la table>)" à partir du nom SQL de la table.

Entrée

- *`strSqlName`* : Nom SQL de la table pour laquelle on veut générer une chaîne de description. Si ce paramètre contient un nom SQL non valide, la fonction renvoie un point d'interrogation ("?").

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant génère une chaîne de description pour la table des biens (Nom SQL : amAsset) :

```
AmTableDesc("amAsset")
```

Le résultat est le suivant :

```
Biens (amAsset)
```

AmTaxRate()

Syntaxe API

```
double AmTaxRate(char *strTaxRateName, long lTaxLocId, long tmDate,
double dValue);
```

Syntaxe Basic interne

```
Function AmTaxRate(strTaxRateName As String, lTaxLocId As Long, tmDate
As Date, dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction calcule un taux de taxe en fonction d'un type de taxe, d'une juridiction fiscale et d'une date.

Entrée

- *strTaxRateName* : Ce paramètre contient le nom SQL du type de taxe utilisé pour calculer le taux de taxe.
- *lTaxLocId* : Ce paramètre contient le numéro d'identifiant de la juridiction fiscale concernée par le type de taxe.
- *tmDate* : Ce paramètre contient la date à laquelle vous souhaitez évaluer le taux de taxe.
- *dValue* : ?

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

AmUpdateDetail()

Syntaxe API

```
long AmUpdateDetail(char *strFieldName);
```

Syntaxe Basic interne

```
Function AmUpdateDetail(strFieldName As String, varValue As Variant) As
Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	

Description

Cette fonction est utilisée dans les assistants d'aide à la saisie. La définition du contexte (la table pour laquelle un enregistrement est mis à jour ou renseigné à l'aide de l'assistant) n'est donc pas nécessaire. La

fonction met à jour un ou renseigne des champs ou des liens du contexte en fonction d'une valeur.

Entrée

- *strFieldName* : Ce paramètre contient le nom SQL du champ à mettre à jour.
- *varValue* : Ce paramètre contient la nouvelle valeur du champ.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmUpdateLoginSlot()

Syntaxe API

```
long AmUpdateLoginSlot(long hApiCnxBase);
```

Syntaxe Basic interne

```
Function AmUpdateLoginSlot() As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	

	Utilisable
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction force la mise à jour des informations relatives au jeton de connexion de l'utilisateur connecté.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmUpdateRecord()

Syntaxe API

```
long AmUpdateRecord(long hApiRecord);
```

Syntaxe Basic interne

```
Function AmUpdateRecord(hApiRecord As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	X
Script d'un assistant	
Script FINISH.DO d'un assistant	X
APIs AssetCenter Web	X

Description

Cette fonction permet de mettre à jour un enregistrement.

Entrée

- `hApiRecord` : Ce paramètre contient un handle sur l'enregistrement que vous souhaitez mettre à jour.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmwAdminLogout()

Syntaxe API

```
long AmwAdminLogout();
```

Syntaxe Basic interne

```
Function AmwAdminLogout() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Termine une session où l'utilisateur était connecté comme administrateur Web.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
iRc = AmwAdminLogout()
```

rem session is closed and nothing else can be done in script section

AmwApproveRequest()

Syntaxe API

```
long AmwApproveRequest(long lReqId, char *strComment, long bCheckCompo,
long bCheckCost, double dMaxCost);
```

Syntaxe Basic interne

```
Function AmwApproveRequest(lReqId As Long, strComment As String,
bCheckCompo As Long, bCheckCost As Long, dMaxCost As Double) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Effectue la validation d'une demande d'achat identifiée par son numéro d'identifiant.

Entrée

- *lReqId* : Identifiant de la demande d'achat (table "amReqApprLine").
- *strComment* : Commentaire du validant.
- *bCheckCompo* : Si ce paramètre a pour valeur "1", la validation sera annulée si la composition de la demande d'achat est changée.
- *bCheckCost* : Si ce paramètre a pour valeur "1", la validation de la demande d'achat est annulée si le montant total de la demande excède le montant spécifié dans le paramètre *dMaxCost*.
- *dMaxCost* : Montant maximal de la demande pour approbation. Ce paramètre n'est utilisé que si *bCheckCost* a pour valeur "1".

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction AmwErrorMsg).

Exemple

```
If (AmwGetTpl("Action") = "Approve") Then
   AmwSetTpl "bApproved", "True"
   iRc = AmApproveRequest(lReqId, strComment, True, False, 0.0)
```

```
Else
  AmwSetTpl "bApproved", "False"
  iRc = AmRefuseRequest(lReqId, strComment)
End If
```

La variable détermine si l'utilisateur veut approuver ou refuser la demande courante. Les variables sont supposées définies dans un autre modèle HTML. Le programme source complet est disponible dans le document rqappr_s.htm, qui fait partie des modèles standard situés dans le sous dossier "websrv\htdocs\tpl\" du dossier d'installation d'AssetCenter.

AmwCloseSlave()

Syntaxe API

```
long AmwCloseSlave(char *strId);
```

Syntaxe Basic interne

```
Function AmwCloseSlave(strId As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Force la déconnexion d'un utilisateur identifié par sa clé de sécurité.

Entrée

- *strId* : Chaîne d'Identification fournie par la variable $$(id). Cette variable est créé automatiquement pour chaque modèle. Sa valeur est celle de la clé de sécurité créée à la connexion à la base de données.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
SlaveId="3B24_4CA70BBD_2A50"
```

IRc = AmwCloseSlave(SlaveId)

AmwCurrentCnx()

Syntaxe API

```
long AmwCurrentCnx();
```

Syntaxe Basic interne

```
Function AmwCurrentCnx() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'identifiant de la connexion sous la forme d'un "Long". Cet identifiant peut être également utilisé pour les APIs AssetCenter.

Sortie

- 0 : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).
- Identifiant de la connexion courante.

Exemple

```
lCnx = AmwCurrentCnx()
print "<H1> Connexion courante : ";lCnx;"</H1>"
```

AmwCurrentUserId()

Syntaxe API

```
long AmwCurrentUserId();
```

Syntaxe Basic interne

```
Function AmwCurrentUserId() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'identifant de l'utilisateur (identifiant d'un enregistrement de la table des services et personnes) de la session courante, tel qu'il est stocké dans la base de données AssetCenter.

Les utilisateurs possédant un login valide sur la base de données sont identifiées par une clé unique. La valeur de cette clé est clle du champ "lEmplDeptId" de la table "amEmplDept".

Sortie

Identifiant de l'utilisateur.

Exemple

```
lUserId = AmwCurrentUserId()
```

print "<H1> Identifiant de l'utilisateur courant :";lUserId;"</H1>"

AmwErrorMsg()

Syntaxe API

```
long AmwErrorMsg(char *return, long lreturn);
```

Syntaxe Basic interne

```
Function AmwErrorMsg() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le dernier message d'erreur.

Sortie

Description de l'erreur survenue au cours de la dernière opération.

Exemple

```
iRc = AmwLogin(DbName,User,password)
```

If iRc <> 0 then print "Error:";AmwErrorMsg()

AmwExist()

Syntaxe API

```
long AmwExist();
```

Syntaxe Basic interne

```
Function AmwExist() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Teste l'existence d'une variable donnée dans les modèles.

Sortie

- False : La variable n'existe pas.
- True : La variable existe.

Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
Rem Set a template variable
AmwSetTpl "MyVar","Some value"
If AmwExist("Myvar") then print "MyVar=";AmwGetTpl("MyVar")
```

AmwFetchQuery()

Syntaxe API

```
long AmwFetchQuery(char *strQuery, long iStart, long iNb);
```

Syntaxe Basic interne

```
Function AmwFetchQuery(strQuery As String, iStart As Long, iNb As Long)
As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Effectue une requête sur la base de données AssetCenter et renvoie *iNb* éléments à partir de la position *iStart*. La première position est "0".

Les résultats sont stockés dans l'espace de données des modèles en utilisant des variables automatiquement indexées. Les champs sur lesquels porte la requête sont utilisés dans le nom de ces variables.

Par exemple, si la requête porte sur le champ "Marque" et que 10 éléments de la base de données sont renvoyés par la requête, les variables "Marque0", ..., "Marque9" seront créées avec les valeurs correspondantes.

Entrée

- *strQuery* : Requête AQL valide.
- *iStart* : Position du premier élément à récupérer (la première position est "0").
- *iNb* : Nombre d'éléments à récupérer.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
iRc = AmwFetchQuery("select Name,Tel from amEmplDept",0,10
</script>
```

Name0 = $$(Name0) Tel0= $$(Tel0)

Name1 = $$(Name1) Tel1= $$(Tel1)

Etc...

Dans cet exemple, les champs "Nom" et "Tel." sont récupérés pour les 10 premiers enregistrements de la table des services et personnes.

Après exécution, dix variables "Nom0", ", "Nom9" ainsi que 10 variables "Tel.0", ..., "Tel.9" sont créées dans l'espace de données des modèles.

AmwGetEnv()

Syntaxe API

```
long AmwGetEnv(char *return, long lreturn, char *strKeyName);
```

Syntaxe Basic interne

```
Function AmwGetEnv(strKeyName As String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	

	Utilisable
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie sous la forme d'une chaîne le contenu de la variable globale d'environnement.

Entrée

- *`strKeyName`* : Nom de la variable.

Sortie

- Chaîne de caractères représentant le contenu de la variable globale d'environnement.
- Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
MyVal = AmwGetEnv("aGlobalVarName")
```

Print "Global value for aGlobalVarName";MyVal

AmwGetTpl()

Syntaxe API

```
long AmwGetTpl(char *return, long lreturn);
```

Syntaxe Basic interne

```
Function AmwGetTpl() As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie sous la forme d'une chaîne le contenu de la variable d'environnement du modèle local.

Sortie

- Chaîne de caractères représentant le contenu de la variable d'environnement du modèle local.
- Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
MyVal = AmwGetTpl("aLocalVarName")
```

```
Print "Local value for aLocVarName";MyVal
```

AmwInclude()

Syntaxe API

```
long AmwInclude();
```

Syntaxe Basic interne

```
Function AmwInclude() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Lit un fichier identifié par son nom et le colle à partir de la position courante du modèle.

$$(AmwInclude("file.htm")) constitue une façon simple de réutiliser des portions de code HTML communes à plusieurs modèles.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<html>
<!"Use standard menubar -->
$$(AmwInclude("menubar.htm"))
```

AmWizChain()

Syntaxe API

```
long AmWizChain(char *strWizSqlName);
```

Syntaxe Basic interne

```
Function AmWizChain(strWizSqlName As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	

Description

Cette fonction exécute un assistant B au sein d'un assistant A. En fin d'exécution de l'assistant B, la fonction rend la main à l'assistant A.

Entrée

- `strWizSqlName` : Nom SQL de l'assistant à exécuter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

AmwLoadFile()

Syntaxe API

```
long AmwLoadFile();
```

Syntaxe Basic interne

```
Function AmwLoadFile() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Annule l'évaluation du modèle courant et utilise un fichier à la place. L'exécution de la section de code courante est achevée et le nouveau document est ensuite utilisé.

Le nouveau document est traité dans le même espace de données. Les variables précédemment définies ne sont pas détruites.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
iRc = AmwFetchQuery("invalid query",0,10)
if iRc <>0 then
  rem error occurred use standard error reporting doc
  iRc = AmwLoadFile("error.htm")
end if
</script>
```

AmwLogin()

Syntaxe API

```
long AmwLogin(char *strDataBase, char *strUserName, char *strPassword);
```

Syntaxe Basic interne

```
Function AmwLogin(strDataBase As String, strUserName As String,
strPassword As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Réalise une connexion sur une base de données AssetCenter. AssetCenter Web utilise les mêmes informations de connexion qu'AssetCenter.

Entrée

- *strDataBase* : Nom de la base de données, tel qu'il apparaît dans l'écran de connexion d'AssetCenter.
- *strUserName* : Login utilisé pour la connexion de l'utilisateur. User login name (SQL name: UserLogin).
- *strPassword* : Mot de passe associé au login. Password (SQL name: lPasswordId).

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction AmwErrorMsg).

Exemple

```
iRc = AmwLogin(DbName,User,Password)
```

If iRc <> 0 then Print "Login failed: ";AmwErrorMsg()

AmwLogout()

Syntaxe API

```
long AmwLogout();
```

Syntaxe Basic interne

```
Function AmwLogout() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Réalise une déconnexion à la base de données. Cette opération met fin à la session de travail courante. Aucune opération de scripting ne doit être effectué après l'appel de cette fonction.

Cette fonction est en général utilisée à la fin de la section de script du modèle de déconnexion.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
 iRc = AmwLogout()
</script>

<H1> Thank you for using this software </H1>
```

AmwNewRecord()

Syntaxe API

```
long AmwNewRecord(char *strTable, char *strFieldList);
```

Syntaxe Basic interne

```
Function AmwNewRecord(strTable As String, strFieldList As String) As
Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Crée un enregistrement vide dans une table AssetCenter identifiée par son nom SQL puis utilise le paramètre *strFieldList* pour définir la valeur initiale des champs de l'enregistrement.

Les valeurs initiales des champs sont lues à partir des variables spécifiées dans le paramètre *strFieldList*. Le paramètre *strFieldList* possède la syntaxe suivante :

```
<Nom SQL du champ 1>(<Nom de la variable>);<Nom SQL du champ 2>(<Nom de la variable>)...
```

Les champs sont mis à jour avec la valeur contenue dans la variable spécifiée entre parenthèses.

Cette fonction créée une variable "<table_name>_id" contenant l'identifiant de l'enregistrement créé.

Entrée

- *strTable* : Nom SQL de la table.
- *strFieldList* : n-uplet de valeurs <Nom SQL du champ>(<Nom de la variable>).

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
AmwSetTpl "Telephone","12121212"
```

AmwSetTpl "UserName","joe"

iRc = AmNewRecord("amEmplDeptId","Name(UserName);Tel(Telephone)")

AmWorkTimeSpanBetween()

Syntaxe API

```
long AmWorkTimeSpanBetween(char *strCalendarSqlName, long tmEnd, long tmStart);
```

Syntaxe Basic interne

```
Function AmWorkTimeSpanBetween(strCalendarSqlName As String, tmEnd As Date, tmStart As Date) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	
AssetCenter APIs	X
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la durée ouvrée entre deux dates. Cette durée est exprimée en secondes et respecte les informations d'un calendrier des périodes ouvrées.

Entrée

- *strCalendarSqlName* : Ce paramètre contient le nom SQL du calendrier des périodes ouvrées utilisé comme base pour le calcul de la durée ouvrée écoulée entre les deux dates. Si ce paramètre est omis, la durée calculée ne tient compte d'aucune période ouvrée.
- *tmEnd* : Ce paramètre contient la date de fin de la période sur laquelle on effectue le calcul de la durée ouvrée.
- *tmStart* : Ce paramètre contient la date de début de la période sur laquelle on effectue le calcul de la durée ouvrée.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant calcule la durée ouvrée entre le 01/09/1998 à 8h00 et le 24/09/1998 à 19h00. Le calendrier utilisé, de nom SQL "Calendar_Paris" définit les périodes ouvrées suivantes :

- Du lundi au jeudi de 8h00 à 12h00, puis de 14h00 à 18h00.
- Le vendredi de 8h00 à 12h00, puis de 14h00 à 17h00.

```
AmWorkTimeSpanBetween("Calendar_Paris", "1998/09/24 19:00:00",
"1998/09/01 08:00:00")
```

Cet exemple renvoie la valeur 507600 qui représente le nombre de secondes ouvrées écoulées entre les deux dates.

AmwOutputIf()

Syntaxe API

```
long AmwOutputIf();
```

Syntaxe Basic interne

```
Function AmwOutputIf() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	

	Utilisable
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Permet la génération du modèle courant en fonction de la valeur d'une variable identifiée par son nom.

La génération est possible si la variable existe et n'est pas nulle. Dans le cas contraire, les lignes de code suivantes sont ignorées jusqu'à ce qu'une autre fonction `AmwOutputIf` soit exécutée avec succès.

Cette fonction peut être utilisée pour effectuer la composition de documents HTML contenant des informations à générer en cas d'échec et en cas d'erreur.

Par exemple, à son exécution, la section de script définit la valeur d'une variable si l'opération a été effectéue avec succès. Cette variable est ensuite utilisée pour déterminer quelle partie du document HTML doit être affichée.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
 rem set one valid var
 AmwSetTpl "GoodVar","true"
 rem set one false var
 AmwSetTpl "FalseVar","False"
```

```
</script>
$$(AmwOutputIf("NoVarsLikeThis"))
This line is not visible because var does not exist<br>
$$(AmwOutputIf("FalseVar"))
This line is not visible because var evaluates as false<br>
$$(AmwOutputIf("GoodVar"))
This line is visible because this var evaluates as true
```

AmwOutputIfNot()

Syntaxe API

```
long AmwOutputIfNot();
```

Syntaxe Basic interne

```
Function AmwOutputIfNot() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	

	Utilisable
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Permet la génération du modèle courant en fonction de la valeur d'une variable identifiée par son nom. Cette fonction est l'exact opposé de la fonction `AmwOutputIf`. La génération est possible si la variable n'existe pas ou est nulle.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

AmwPlusToSpace()

Syntaxe API

```
long AmwPlusToSpace(char *return, long lreturn, char *strVal);
```

Syntaxe Basic interne

```
Function AmwPlusToSpace(strVal As String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Decode une chaîne de type CGI en une chaîne de caractères affichables.

Cette fonction peut être utilisée pour décoder manuellement les chaînes fournies par les mécanismes CGI.

Entrée

- *strVal* : Chaîne à décoder.

Sortie

- Chaîne de caractères décodée.
- Le message d'erreur est accessible au travers de la fonction AmwErrorMsg.

Exemple

```
<script Language="amScript1.0">
 varText = "%3F+Cgi+like+string+%3F"
 AmwSetTpl "param1",AmwPlusToSpace(varText)
```

```
</script>

Following line should display: ? cgi like string ?
$$(Param1)
```

AmwPrintDebug()

Syntaxe API

```
void AmwPrintDebug(char *strVal);
```

Syntaxe Basic interne

```
Function AmwPrintDebug(strVal As String)
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Affiche une chaîne dans la fenêtre de console locale. Cette fonction ne peut être utilisée que si AssetCenter Web a été lancé dans sa version exécutable.

Quand AssetCenter Web fonctionne sous la forme d'un service, les informations de débogage ne peuvent être envoyées qu'à un utilisateur connecté. Si `amw3.exe` est démarré manuellement en double-cliquant sur l'icône du programme, une fenêtre de console est créée, qui affiche des informations complémentaires.

Cette fonction permet d'éviter de surcharger les documents HTML générés avec des informations de débogage.

Entrée

- *`strVal`* : Message à afficher.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
  Print "This text goes to the client"
  AmwPrintDebug "This Text goes in the local console window"
</script>
```

AmwQStrDelKey()

Syntaxe API

```
long AmwQStrDelKey(char *return, long lreturn, char *strQStr, char
*strKey);
```

Syntaxe Basic interne

```
Function AmwQStrDelKey(strQStr As String, strKey As String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Détruit une clé donnée dans une requête CGI.

Cette fonction permet de manipuler facilement les requêtes et évite l'analyse "manuelle" de la requête.

Entrée

- *strQStr* : Requête originale sous la forme d'une chaîne de caractères.
- *strKey* : Nom de la clé à détruire.

Sortie

- Requête mise à jour.
- Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
<script Language="AmScript1.0">
 rem get current query string and remove name parameter
 AmwSetTpl "NewQueryStr",AmwQStrDelKey(T_Query,"Name")
</script>

<A HREF="doc.htm?$$(NewQueryStr)"> Submit with no name </A>
```

AmwQStrSetKey()

Syntaxe API

```
long AmwQStrSetKey(char *return, long lreturn, char *strQStr, char
*strKey, char *strVal);
```

Syntaxe Basic interne

```
Function AmwQStrSetKey(strQStr As String, strKey As String, strVal As
String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Modifie ou crée un couple clé/ valeur dans une requête CGI.

Entrée

- `strQStr` : Requête originale sous la forme d'une chaîne de caractères.
- `strKey` : Nom de la clé à modifier ou à ajouter.
- `strVal` : Valeur de la clé.

Sortie

- Requête mise à jour.
- Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
<script Language="AmScript1.0">
 rem get current query string an change name parameter
 AmwSetTpl "NewQueryStr",AmwQStrSetKey(T_Query,"Name","joe")
</script>

<A HREF="doc.htm?$$(NewQueryStr)"> Submit with joe name </A>
```

AmwRefuseRequest()

Syntaxe API

```
long AmwRefuseRequest(long lReqId, char *strComment);
```

Syntaxe Basic interne

```
Function AmwRefuseRequest(lReqId As Long, strComment As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Refuse une demande d'achat, identifiée par son identifiant *lReqId*.

Entrée

- *lReqId* : Identifiant de la demande d'achat (table "amReqApprLine").
- *strComment* : Commentaire du validant.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction AmwErrorMsg).

AmwSetEnv()

Syntaxe API

```
void AmwSetEnv(char *strKeyName, char *strValue);
```

Syntaxe Basic interne

```
Function AmwSetEnv(strKeyName As String, strValue As String)
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Définit ou crée une variable possédant une valeur définie dans l'espace des variables de la session globale.

Entrée

- *strKeyName* : Nom de la variable.
- *strValue* : Valeur de la variable sous la forme d'une chaîne.

Sortie

Le message d'erreur est accessible au travers de la fonction AmwErrorMsg.

Exemple

```
<script Language="AmScript1.0">
 rem Set a variable for the whole session
 AmwSetEnv "MyRealName","Joe"
```

</script>

AmwSetTpl()

Syntaxe API

```
void AmwSetTpl();
```

Syntaxe Basic interne

```
Function AmwSetTpl()
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Définit ou crée une variable possédant une valeur définie dans l'espace des variables du modèle local.

Sortie

Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
<script Language="AmScript1.0">
 rem Set a variable local to this template and visible in html
 AmwSetEnv "StockValue","35$"
</script>

Value: $$(StockValue)<br>
```

AmwSetWebAdminPwd()

Syntaxe API

```
long AmwSetWebAdminPwd(char *strNew, char *strOld);
```

Syntaxe Basic interne

```
Function AmwSetWebAdminPwd(strNew As String, strOld As String) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Change le mot de passe de l'administrateur Web.

Entrée

- `strNew` : Nouveau mot de passe.
- `strOld` : Ancien mot de passe.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<script Language="AmScript1.0">
 rem Change password in admin session
 iRc =  AmwSetWebAdminPwd("","K7TR89M")
</script>
```

AmwSetWebTimeout()

Syntaxe API

```
long AmwSetWebTimeout(long iTimeout);
```

Syntaxe Basic interne

```
Function AmwSetWebTimeout(iTimeout As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Change la valeur par défaut de la minuterie de déconnexion pour les sessions utilisateur définies dans `amw3.ini`.

Si aucune activité n'est enregistrée pendant *iTimeOut* minutes l'utilisateur est déconnecté.

Entrée

- *iTimeout* : Nombre de minutes avant déconnexion d'un utilisateur inactif.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction AmwErrorMsg).

Exemple

```
<script Language="AmScript1.0">
 rem Change timeout value to 3 mn
 iRc =  AmwSetWebTimeOut(3)
</script>
```

AmwSlaveList()

Syntaxe API

```
long AmwSlaveList();
```

Syntaxe Basic interne

```
Function AmwSlaveList() As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Affiche la liste de toutes les connexions actives.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction `AmwErrorMsg`).

Exemple

```
<H1> Active users:</H1>
```

$$(AmwSlaveList())

AmwSpaceToPlus()

Syntaxe API

```
long AmwSpaceToPlus(char *return, long lreturn, char *strVal);
```

Syntaxe Basic interne

```
Function AmwSpaceToPlus(strVal As String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Encode une chaîne de caractères ASCII de manière à la rendre compatible avec les chaînes de type CGI.

Cette fonction est utile pour créer des liens quand des informations complémentaires doivent être transmises au modèle destinataire.

Entrée

- *strVal* : Chaîne à encoder.

Sortie

- Chaîne encodée.
- Le message d'erreur est accessible au travers de la fonction AmwErrorMsg.

Exemple

```
<script Language="amScript1.0">
 varText = "Name with strange chars +=$? "
 AmwSetTpl "param1",AmwPlusToSpace(varText)
</script>

<A HREF="doc.htm?$$Value=$$(param1)> Click me </A>
```

AmwStrReplace()

Syntaxe API

```
long AmwStrReplace(char *return, long lreturn, char *strString, char
*strFind, char *strRepl);
```

Syntaxe Basic interne

```
Function AmwStrReplace(strString As String, strFind As String, strRepl
As String) As String
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Remplace toutes les occurrences de la chaîne *`strFind`* dans la chaîne *`strString`* par la chaîne *`strRepl`*.

Entrée

- *`strString`* : Chaîne à traîter.
- *`strFind`* : Chaîne à remplacer.
- *`strRepl`* : Chaîne de remplacement.

Sortie

- Chaîne après remplacement.
- Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
<script Language="AmScript1.0">
  rem replace all occurrences of joe with jack
  resStr = AmwStrReplace("Default user is joe","joe","jack")
</script>
```

AmwUpdateRecord()

Syntaxe API

```
long AmwUpdateRecord(char *strTable, char *strFieldList, long lId);
```

Syntaxe Basic interne

```
Function AmwUpdateRecord(strTable As String, strFieldList As String,
lId As Long) As Long
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	X

Description

Met à jour les champs d'un enregistrement. Celui-ci est identifié par son identifiant et le nom SQL de la table dans laquelle il se trouve.

Les nouvelles valeurs des champs sont lues à partir des variables spécifiées dans le paramètre *strFieldList*. Le paramètre *strFieldList* possède la syntaxe suivante :

<Nom SQL du champ 1>(<Nom de la variable>);<Nom SQL du champ 2>(<Nom de la variable>)...

Les champs sont mis à jour avec la valeur contenue dans la variable spécifiée entre parenthèses.

Entrée

- *strTable* : Nom de la table contenant l'enregistrement à modifier.
- *strFieldList* : n-uplet de valeurs <Nom SQL du champ>(<Nom de la variable>).
- *lId* : Identifiant de l'enregistrement.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur (accessible au travers de la fonction AmwErrorMsg).

Exemple

```
<script Language="AmScript1.0">
```

AmwSetTpl "NewTel","12121212"

AmwSetTpl "NewName","joe"

iRc=AmUpdateRecord("amEmplDeptId","Name(NewName);Tel(NewTel)",lId)

</script>

AmwVars()

Syntaxe API

```
void AmwVars();
```

Syntaxe Basic interne

```
Function AmwVars()
```

Champ d'application

Version : 2.52

	Utilisable
Fonction Builtin	
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	
Script de définition d'un paramètre de caractéristique	
Action de type "Script"	
Script d'un assistant	
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Affiche le contenu des variables locales et globales dans le document courant.

Sortie

Le message d'erreur est accessible au travers de la fonction `AmwErrorMsg`.

Exemple

```
<H1>Available variables:</H1>
```

$$(AmwVars())

AppendOperand()

Syntaxe Basic interne

```
Function AppendOperand(strExpr As String, strOperator As String,
strOperand As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Concatène une chaîne en fonction des paramètres passés à la fonction. Le résultat a la forme suivante :

strExpr strOperator strOperand

Entrée

- *strExpr* : Expression à concaténer.
- *strOperator* : Opérateur à concaténer.
- *strOperand* : Opérande à concaténer.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Si l'un des paramètres *strExpr* ou *strOperand* est omis, *strOperator* n'est pas utilisé dans la concaténation.

ApplyNewVals()

Syntaxe Basic interne

```
Function ApplyNewVals(strValues As String, strNewVals As String,
strRows As String, strRowFormat As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Affecte des valeurs identiques pour les cellules identifiées d'un contrôle "ListBox".

Entrée

- *strValues* : Chaîne source contenant les valeurs d'un contrôle "ListBox" à traiter.

- *strNewVals* : Nouvelle valeur à affecter aux cellules concernées.
- *strRows* : Identifiants des lignes à traiter. Les identifiants sont séparés par une virgule.
- *strRowFormat* : Instructions de formatage de la sous-liste. Les instructions sont séparées par le caractère "|". Chaque instruction représente le numéro de la colonne qui contiendra *strNewVals*

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Asc()

Syntaxe Basic interne

```
Function Asc(strAsc As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le code ASCII du premier caractère d'une chaîne.

Entrée

- `strAsc` : Chaîne de caractères sur laquelle opère la fonction.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Sub Main ()
   Dim I, Msg                  ' Declare variables.
   For I = Asc("A") To Asc("Z")    ' From A through Z.
   Msg = Msg & Chr(I)          ' Create a string.
   Next I
   Print Msg                   ' Display results.
End Sub
```

Atn()

Syntaxe Basic interne

```
Function Atn(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'arc tangente d'un nombre, exprimé en radians

Entrée

- `dValue` : Nombre dont vous souhaitez connaître l'arc tangente.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub AtnExample ()
   Dim Msg, Pi       ' Declare variables.
   Pi = 4 * Atn(1)   ' Calculate Pi.
   Msg = "Pi is equal to " & Str(Pi)
   Print Msg         ' Display results.
End Sub
```

BasicToLocalDate()

Syntaxe Basic interne

```
Function BasicToLocalDate(strDateBasic As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format Basic en une date au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows).

Entrée

- *`strDateBasic`* : Date au format Basic à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

BasicToLocalTime()

Syntaxe Basic interne

```
Function BasicToLocalTime(strTimeBasic As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une heure au format Basic en une heure au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows).

Entrée

- `strTimeBasic` : Heure au format Basic à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

BasicToLocalTimeStamp()

Syntaxe Basic interne

```
Function BasicToLocalTimeStamp(strTSBasic As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit un ensemble Date+Heure au format Basic en un ensemble Date+Heure au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows).

Entrée

- `strTSBasic` : Date+Heure au format Basic à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Beep()

Syntaxe Basic interne

```
Function Beep()
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Emet un son (un beep) sur la machine.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

CDbl()

Syntaxe Basic interne

```
Function CDbl(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expression en un double ("Double").

Entrée

- *dValue* : Expression à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
   Dim y As Long

   y = 25
   If  VarType(y) = 2 Then
   Print y
   x = CDbl(y)   'Converts the long value of y to an integer value in x
      Print x
   End If
End Sub
```

ChDir()

Syntaxe Basic interne

```
Function ChDir(strDirectory As String)
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Change le répertoire courant.

Entrée

- `strDirectory` : Nouveau répertoire courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

ChDrive()

Syntaxe Basic interne

```
Function ChDrive(strDrive As String)
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Change le lecteur courant.

Entrée

- *strDrive* : Nouveau lecteur.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Chr()

Syntaxe Basic interne

```
Function Chr(iChr As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la chaîne correspondant au code ASCII passé par le paramètre *iChr*.

Entrée

- *iChr* : Code ASCII du caractère.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub ChrExample ()
  Dim X, Y, Msg, NL
  NL = Chr(10)
  For X = 1 to 2
    For Y = Asc("A") To Asc("Z")
      Msg = Msg & Chr(Y)
    Next Y
```

```
    Msg = Msg & NL
    Next X
    Print Msg
End Sub
```

CInt()

Syntaxe Basic interne

```
Function CInt(iValue As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expressions en un entier ("Integer").

Entrée

- *iValue* : Expression à convertir.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Sub Main ()
   Dim y As Long

   y = 25
   If  VarType(y) = 2 Then
   Print y
   x = CInt(y)   'Converts the long value of y to an integer value in x
      Print x
   End If
End Sub
```

CLng()

Syntaxe Basic interne

```
Function CLng(lValue As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expression en un long ("Long").

Entrée

- `lValue` : Expression à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
  Dim y As Integer
```

```
    y = 25
    If  VarType(y) = 2 Then
    Print y
    x = CLng(y)   'Converts the integer value of y to a long value in x
      Print x
    End If
End Sub
```

Cos()

Syntaxe Basic interne

```
Function Cos(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	X

Description

Renvoie le cosinus d'un nombre, exprimé en radians.

Entrée

- *dValue* : Nombre dont vous souhaitez connaître le cosinus.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

CountOccurences()

Syntaxe Basic interne

```
Function CountOccurences(strSearched As String, strPattern As String,
strEscChar As String) As Long
```

Champ d'application

Version : 3.5

	Utilisable

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Compte le nombre d'occurrences d'une chaîne de caractères à l'intérieur d'une autre chaîne.

Entrée

- `strSearched` : Chaîne de caractères à l'intérieur de laquelle s'effectue la recherche.
- `strPattern` : Chaîne de caractères à rechercher à l'intérieur de `strSearched`.
- `strEscChar` : Caractère d'échappement. Si la fonction rencontre ce caractère à l'intérieur de la chaîne `strSearched`, la recherche s'arrête.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=CountOccurences("toi|moi|toi,moi|toi", "toi", ",") 'Renvoie la valeur "2"

MyStr=CountOccurences("toi|moi|toi,moi|toi", "toi", "|") 'Renvoie la valeur "1"

CountValues()

Syntaxe Basic interne

```
Function CountValues(strSearched As String, strSeparator As String,
strEscChar As String) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Compte le nombre d'éléments dans une chaîne de caractères en tenant compte d'un séparateur et d'un caractère d'échappement.

Entrée

- *strSearched* : Chaîne de caractères à traiter.
- *strSeparator* : Séparateur utilisé pour délimiter les éléments.
- *strEscChar* : Caractère d'échappement. Si ce caractère préfixe un séparateur, ce dernier sera ignoré.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=CountValues("toi|moi|toi\|moi|toi", "|", "\") 'Renvoie la valeur 4

MyStr=CountValues("toi|moi|toi\|moi|toi", "|", "") 'Renvoie la valeur 5

CSng()

Syntaxe Basic interne

```
Function CSng(fValue As Single) As Single
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expression en un nombre à virgule flottante ("Float").

Entrée

- `fValue` : Expression à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

CStr()

Syntaxe Basic interne

```
Function CStr(strValue As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expression en une chaîne de caractères ("String").

Entrée

- *strValue* : Expression à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

CurDir()

Syntaxe Basic interne

```
Function CurDir() As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le chemin courant.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

CVar()

Syntaxe Basic interne

```
Function CVar(vValue As Variant) As Variant
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une expression en un variant ("Variant").

Entrée

- `vValue` : Expression à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Date()

Syntaxe Basic interne

```
Function Date() As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la date courante du système.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

DateSerial()

Syntaxe Basic interne

```
Function DateSerial(iYear As Long, iMonth As Long, iDay As Long) As
Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie une date formatée en fonction des paramètres *iYear*, *iMonth* et *iDay*.

Entrée

- *iYear* : Année. Si sa valeur est comprise entre 0 et 99, ce paramètre décrit les années de 1900 à 1999. Pour toutes les autres années, vous devez utiliser un nombre de quatre chiffres (par exemple 1800).
- *iMonth* : Mois.
- *iDay* : Jour.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Chacun de ces paramètres peut prendre pour valeur une expression numérique représentant un nombre de jours, de mois ou d'années. Ainsi l'exemple suivant :

```
DateSerial(1999-10, 3-2, 15-8)
```

renvoie la valeur :

```
1989/1/7
```

Lorsque la valeur d'un paramètre est en dehors de l'intervalle de valeurs généralement admis (c'est à dire 1-31 pour les jours, 1-12 pour les mois, …), elle est convertie vers le paramètre immédiatement supérieur. Ainsi, si vous entrez "35" comme valeur pour le paramètre iDay, ce dernier sera interprété comme 1 mois et un jour.

L'exemple suivant :

```
DateSerial (1999-50, 9-5, 1-2)
```

renvoie la valeur :

```
1949/3/30
```

DateValue()

Syntaxe Basic interne

```
Function DateValue(tmDate As Date) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la partie date d'une valeur "Date+Heure"

Entrée

- *tmDate* : Date au format "Date+Heure".

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant :

```
DateValue ("1999/09/24 15:00:00")
```

renvoie la valeur :

```
1999/09/24
```

Day()

Syntaxe Basic interne

```
Function Day(tmDate As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le jour contenu dans le paramètre *tmDate*.

Entrée

- *tmDate* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Dim MyStr
```

MyStr=Day(AmDate())'Renvoie le jour courant

EscapeSeparators()

Syntaxe Basic interne

```
Function EscapeSeparators(strSource As String, strSeparators As String,
strEscChar As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Préfixe un ou plusieurs caractère(s) défini(s) comme séparateur(s) par un caractère d'échappement.

Entrée

- *strSource* : Chaîne de caractères à traiter.

- *strSeparators* : Liste des séparateurs à préfixer. Si vous souhaitez déclarer plusieurs séparateurs, vous devez les séparer par le caractère utilisé comme caractère d'échappement (indiqué dans le paramètre *strEscChar*.
- *strEscChar* : Caractère d'échappement. Il préfixera tous les séparateurs définis dans *strSeparators*

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=EscapeSeparators("toi|moi|toi,moi|toi", "|\,", ",") 'Renvoie la valeur "toi\|moi\|toi\,moi\|toi"

Exp()

Syntaxe Basic interne

```
Function Exp(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'exponentielle d'un nombre.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître l'exponentielle.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

ExtractValue()

Syntaxe Basic interne

```
Function ExtractValue(strSeparator As String, strEscChar As String) As
String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait d'une chaîne de caractères les valeurs délimitées par un séparateur. La valeur récupérée est alors effacée de la chaîne source. Cette opération tient compte d'un éventuel caractère d'échappement. Si le séparateur n'est pas trouvé dans la chaîne source, l'intégralité de la chaîne est renvoyée et la chaîne source est entièrement effacée.

Entrée

- *strSeparator* : Caractère utilisé comme séparateur dans la chaîne source.
- *strEscChar* : Caractère d'échappement. Si ce caractère préfixe le séparateur, ce dernier est ignoré.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=ExtractValue("toi,moi", ",", "\") 'Renvoie "toi" et laisse "moi" dans la chaîne source

MyStr=ExtractValue(",toi,moi", ",", "\") 'Renvoie "" et laisse "toi,moi" dans la chaîne source

MyStr=ExtractValue("toi", ",", "\") 'Renvoie "toi" et laisse "" dans la chaîne source

MyStr=ExtractValue("toi\,moi", ",", "\") 'Renvoie "toi\,moi" et laisse "" dans la chaîne source

MyStr=ExtractValue("toi\,moi", ",", "") 'Renvoie "toi\" et laisse "moi" dans la chaîne source

FileCopy()

Syntaxe Basic interne

```
Function FileCopy(strSource As String, strDest As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Copie un fichier ou un répertoire.

Entrée

- `strSource` : Chemin complet du fichier ou du répertoire à copier.
- `strDest` : chemin complet du fichier ou du répertoire de destination.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

FileDateTime()

Syntaxe Basic interne

```
Function FileDateTime(strFileName As String) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la date et l'heure d'un fichier sous la forme d'un "Long".

Entrée

- `strFileName` : Chemin complet du fichier concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

FileLen()

Syntaxe Basic interne

```
Function FileLen(strFileName As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la taille d'un fichier.

Entrée

- *strFileName* : Chemin complet du fichier concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Fix()

Syntaxe Basic interne

```
Function Fix(dValue As Double) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la partie entière (premier entier supérieur dans le cas d'un nombre négatif) d'un nombre à virgule.

Entrée

- *dValue* : Nombre dont vous souhaitez connaître la partie entière.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Format()

Syntaxe Basic interne

```
Function Format(vValue As Variant, strFormat As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Formate un nombre en fonction de l'expression contenue dans le paramètre `strFormat`.

Entrée

- `vValue` : Nombre à formater.
- `strFormat` : Expression contenant les instructions de formatage.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple de code suivant montre comment formater un nombre :

```
Dim MyValue

MyValue=3245.69

MyStr=FormatDate(MyValue, "###0.000") 'Renvoie "Tuesday 14
March 2000"
```

FormatDate()

Syntaxe Basic interne

```
Function FormatDate(tmFormat As Date, strFormat As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Formate une date en fonction de l'expression contenue dans le paramètre *strFormat*.

Entrée

- *tmFormat* : Date à formater.
- *strFormat* : Expression contenant les instructions de formatage.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple de code suivant montre comment formater une date :

```
Dim MyDate

MyDate="2000/03/14"

MyStr=FormatDate(MyDate, "dddd d mmmm yyyy") 'Renvoie "Tuesday
14 March 2000"
```

FormatResString()

Syntaxe Basic interne

```
Function FormatResString(strResString As String, strParamOne As String,
strParamTwo As String, strParamThree As String, strParamFour As String,
strParamFive As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction traite une chaîne source en remplaçant les variables $1, $2, $3, $4 et $5 respectivement par les chaînes contenues dans les paramètres `strParamOne`, `strParamTwo`, `strParamThree`, `strParamFour` et `strParamFive`.

Entrée

- `strResString` : Chaîne source à traiter.
- `strParamOne` : Chaîne de remplacement de la variable $1.
- `strParamTwo` : Chaîne de remplacement de la variable $2.
- `strParamThree` : Chaîne de remplacement de la variable $3.
- `strParamFour` : Chaîne de remplacement de la variable $4.
- `strParamFive` : Chaîne de remplacement de la variable $5.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant :

```
FormatResString("je$1il$2vous$3", "tu", "nous", "ils")
```

renvoie "jetuilnousvousils".

FormatString()

Syntaxe Basic interne

```
Function FormatString(strValue As String, strFormat As String) As
String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Formate une chaîne de caractères en fonction de l'expression contenue dans le paramètre *strFormat*.

Entrée

- *strValue* : Chaîne de caractères à formater.
- *strFormat* : Expression contenant les instructions de formatage.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple de code suivant montre comment formater une chaîne de caractères :

```
Dim MyString

MyString="2000/03/14"

MyStr=FormatDate(MyDate, "dddd d mmmm yyyy") 'Renvoie "Tuesday
14 March 2000"
```

FV()

Syntaxe Basic interne

```
Function FV(dblRate As Double, iNper As Long, dblPmt As Double, dblPV
As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le futur montant d'une annuité basée sur des versements constants et périodiques, et sur un taux d'intérêt fixe.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *iNper* : Ce paramètre contient le nombre total d'échéances de l'opération financière.
- *dblPmt* : Ce paramètre indique le montant du paiement à effectuer à chaque échéance. Le paiement comporte généralement le principal et les intérêts.
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les paramètres *Rate* et *Nper* doivent être calculés à l'aide d'échéances exprimées dans les mêmes unités.
- Les sommes versées (exprimées notamment par le paramètre *Pmt*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

GetListItem()

Syntaxe Basic interne

```
Function GetListItem(strFrom As String, strSep As String, lNb As Long,
strEscChar As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la *lNb*ième portion d'une chaîne délimitée par des séparateurs.

Entrée

- *strFrom* : Chaîne source à traiter.
- *strSep* : Caractère utilisé comme séparateur dans la chaîne source.

- *lNb* : Position de la chaîne à récupérer.
- *strEscChar* : Caractère d'échappement. Si ce caractère préfixe un séparateur, ce dernier sera ignoré.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant :

```
GetListItem("ceci_est_un_test", "_", 2, "%")
```

renvoie "est".

```
GetListItem("ceci%_est_un_test", "_", 2, "%")
```

renvoie "un".

Hex()

Syntaxe Basic interne

```
Function Hex(dValue As Double) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la valeur hexadécimale d'un nombre.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître la valeur hexadécimale.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Hour()

Syntaxe Basic interne

```
Function Hour(tmTime As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la valeur de l'heure contenue dans le paramètre *tmTime*.

Entrée

- *tmTime* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Dim MyStr
```

MyStr=Hour(AmDate())'Renvoie la valeur de l'heure courante par exemple "15" s'il est actuellement 15:45:30

InStr()

Syntaxe Basic interne

```
Function InStr(iPosition As Long, strSource As String, strPattern As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la position de la première occurrence d'une chaîne de caractères à l'intérieur d'une autre chaîne de caractères.

Entrée

- *iPosition* : Position de départ de la recherche. Ce paramètre ne peut être négatif et ne doit pas dépasser 65.535.
- *strSource* : Chaîne dans laquelle s'effectue la recherche.
- *strPattern* : Chaîne de caractères à rechercher.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Sub Main ()
   B$ = "Good Bye"
   A% = InStr(2, B$, "Bye")
   C% = Instr(3, B$, "Bye")
End Sub
```

Int()

Syntaxe Basic interne

```
Function Int(dValue As Double) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la partie entière (premier nombre entier inférieur dans le cas d'un nombre négatif) d'un nombre à virgule.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître la partie entière.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

IPMT()

Syntaxe Basic interne

```
Function IPMT(dblRate As Double, iPer As Long, iNper As Long, dblPV As
Double, dblFV As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le montant des intérêts pour une échéance donnée d'une annuité.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *iPer* : Ce paramètre indique la période concernée par le calcul, comprise entre 1 et la valeur du paramètre *Nper*.
- *iNper* : Ce paramètre contient le nombre total d'échéances de l'opération financière.
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.
- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les paramètres *Rate* et *Nper* doivent être calculés à l'aide d'échéances exprimées dans les mêmes unités.

- Les sommes versées (exprimées notamment par le paramètre *Pmt*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

Kill()

Syntaxe Basic interne

```
Function Kill(strKilledFile As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Efface un fichier.

Entrée

- *strKilledFile* : Chemin complet du fichier concerné par l'opération.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

LCase()

Syntaxe Basic interne

```
Function LCase(strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Passe tous les caractères d'une chaîne en minuscules.

Entrée

- *strString* : Chaîne de caractères à passer en minuscules.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
' This example uses the LTrim and RTrim functions to strip
leading ' and trailing spaces, respectively, from a string
variable.

' It uses the Trim function alone to strip both types of spaces.

' LCase and UCase are also shown in this example as well as the use

' of nested function calls


Sub Main
   MyString = "  <-Trim->  "    ' Initialize string.
   TrimString = LTrim(MyString)    ' TrimString = "<-Trim->  ".
   Print "|" & TrimString & "|"
   TrimString = LCase(RTrim(MyString))    ' TrimString = "  <-trim->".
   Print "|" & TrimString & "|"
   TrimString = LTrim(RTrim(MyString))    ' TrimString = "<-Trim->".
   Print "|" & TrimString & "|"
   ' Using the Trim function alone achieves the same result.
   TrimString = UCase(Trim(MyString))    ' TrimString = "<-TRIM->".
```

```
    Print "|" & TrimString & "|"
End Sub
```

Left()

Syntaxe Basic interne

```
Function Left(strString As String, iNumber As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie les iNumber premiers caractères d'une chaîne en partant de la gauche.

Entrée

- *strString* : Chaîne de caractères à traiter.
- *iNumber* : Nombre de caractères à renvoyer.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
  Dim LWord, Msg, RWord, SpcPos, UsrInp    ' Declare variables.
  Msg = "Enter two words separated by a space."
  UsrInp = InputBox(Msg)    ' Get user input.
  print UsrInp
  SpcPos = InStr(1, UsrInp, " ")    ' Find space.
  If SpcPos Then
    LWord = Left(UsrInp, SpcPos - 1)    ' Get left word.
    print "LWord: ";  LWord
    RWord = Right(UsrInp, Len(UsrInp) - SpcPos)    ' Get right word.
    Msg = "The first word you entered is " & LWord
    Msg = Msg & "." & " The second word is "
    Msg = "The first word you entered is <" & LWord & ">"
    Msg = Msg & RWord & "."
  Else
    Msg = "You didn't enter two words."
  End If
  Print Msg    ' Display message.
```

```
    MidTest = Mid("Mid Word Test", 4, 5)
    Print MidTest
End Sub
```

LeftPart()

Syntaxe Basic interne

```
Function LeftPart(strFrom As String, strSep As String, bCaseSensitive
As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait la portion d'une chaîne de caractères située à gauche du séparateur précisé dans le paramètre *strSep*.

La recherche du séparateur s'effectue de la gauche vers la droite.

La recherche peut tenir compte ou non de la casse en fonction de la valeur du paramètre *bCaseSensitive*.

Entrée

- *strFrom* : Chaîne source à traiter.
- *strSep* : Caractère utilisé comme séparateur dans la chaîne source.
- *bCaseSensitive* : En fonction de la valeur de ce paramètre, la recherche respecte (=1) ou non (=0) la casse.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Ces exemples illustrent l'utilisation des fonctions `LeftPart`, `LeftPartFromRight`, `RightPart` et `RightPartFromLeft` sur une même chaîne de caractères: "Ceci_est_un_test" :

```
LeftPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci".

```
LeftPartFromRight("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci_est_un".

```
RightPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "test".

```
RightPartFromLeft("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "est_un_test".

LeftPartFromRight()

Syntaxe Basic interne

```
Function LeftPartFromRight(strFrom As String, strSep As String,
bCaseSensitive As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait la portion d'une chaîne de caractères située à gauche du séparateur précisé dans le paramètre *strSep*.

La recherche du séparateur s'effectue de la droite vers la gauche.

La recherche peut tenir compte ou non de la casse en fonction de la valeur du paramètre *bCaseSensitive*.

Entrée

- *strFrom* : Chaîne source à traiter.
- *strSep* : Caractère utilisé comme séparateur dans la chaîne source.
- *bCaseSensitive* : En fonction de la valeur de ce paramètre, la recherche respecte (=1) ou non (=0) la casse.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Ces exemples illustrent l'utilisation des fonctions `LeftPart`, `LeftPartFromRight`, `RightPart` et `RightPartFromLeft` sur une même chaîne de caractères: "Ceci_est_un_test" :

```
LeftPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci".

```
LeftPartFromRight("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci_est_un".

```
RightPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "test".

```
RightPartFromLeft("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "est_un_test".

Len()

Syntaxe Basic interne

```
Function Len(vValue As Variant) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le nombre de caractères d'une chaîne ou d'un variant.

Entrée

- `vValue` : Variant concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
   A$ = "Cypress Enable"
   StrLen% = Len(A$)    'the value of StrLen is 14
   Print StrLen%
End Sub
```

LocalToBasicDate()

Syntaxe Basic interne

```
Function LocalToBasicDate(strDateLocal As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une date au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows) en une date au format Basic.

Entrée

- `strDateLocal` : Date au format chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

LocalToBasicTime()

Syntaxe Basic interne

```
Function LocalToBasicTime(strTimeLocal As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit une heure au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows) en une heure au format Basic.

Entrée

- `strTimeLocal` : Heure au format chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

LocalToBasicTimeStamp()

Syntaxe Basic interne

```
Function LocalToBasicTimeStamp(strTSLocal As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction convertit un ensemble Date+Heure au format chaîne (telle qu'elle est affichée dans le "control panel" de Windows) en un ensemble Date+Heure au format Basic.

Entrée

- *strTSLocal* : Date+Heure au format chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Log()

Syntaxe Basic interne

```
Function Log(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le logarithme néperien d'un nombre.

Entrée

- *dValue* : Nombre dont vous souhaitez connaître le logarithme.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

LTrim()

Syntaxe Basic interne

```
Function LTrim(strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Supprime tous les espaces précédant le premier caractère (différent d'un espace) d'une chaîne.

Entrée

- *`strString`* : Chaîne de caractères à traiter.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
' This example uses the LTrim and RTrim functions to strip
leading ' and trailing spaces, respectively, from a string
variable.

' It uses the Trim function alone to strip both types of spaces.

' LCase and UCase are also shown in this example as well as the use

' of nested function calls


Sub Main
```

```
    MyString = "  <-Trim->  "    ' Initialize string.
    TrimString = LTrim(MyString)    ' TrimString = "<-Trim->  ".
    Print "|" & TrimString & "|"
    TrimString = LCase(RTrim(MyString))    ' TrimString = "  <-trim->".
    Print "|" & TrimString & "|"
    TrimString = LTrim(RTrim(MyString))    ' TrimString = "<-Trim->".
    Print "|" & TrimString & "|"
    ' Using the Trim function alone achieves the same result.
    TrimString = UCase(Trim(MyString))    ' TrimString = "<-TRIM->".
    Print "|" & TrimString & "|"
End Sub
```

MakeInvertBool()

Syntaxe Basic interne

```
Function MakeInvertBool(lValue As Long) As Long
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X

	Utilisable
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie un booléen inversé (0 devient 1, tout autre nombre devient 0).

Entrée

- *`lValue`* : Nombre concerné par l'opération.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyValue
MyValue=MakeInvertBool(0) 'Renvoie la valeur 1
MyValue=MakeInvertBool(1) 'Renvoie la valeur 0
MyValue=MakeInvertBool(254) 'Renvoie la valeur 0
```

Mid()

Syntaxe Basic interne

```
Function Mid(strString As String, iStart As Long, iLen As Long) As
String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait une chaîne de caractères contenue dans une autre chaîne.

Entrée

- *strString* : Chaîne de caractères concernée par l'opération.
- *iStart* : Position de départ de la chaîne à extraire à l'intérieur de strString.

- *iLen* : longueur de la chaîne à extraire.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
  Dim LWord, Msg, RWord, SpcPos, UsrInp    ' Declare variables.
  Msg = "Enter two words separated by a space."
  UsrInp = InputBox(Msg)    ' Get user input.
  print UsrInp
  SpcPos = InStr(1, UsrInp, " ")    ' Find space.
  If SpcPos Then
    LWord = Left(UsrInp, SpcPos - 1)    ' Get left word.
    print "LWord: ";  LWord
    RWord = Right(UsrInp, Len(UsrInp) - SpcPos)    ' Get right word.
    Msg = "The first word you entered is " & LWord
    Msg = Msg & "." & " The second word is "
    Msg = "The first word you entered is <" & LWord & ">"
    Msg = Msg & RWord & "."
  Else
    Msg = "You didn't enter two words."
  End If
  Print Msg    ' Display message.
  MidTest = Mid("Mid Word Test", 4, 5)
```

```
    Print MidTest
End Sub
```

Minute()

Syntaxe Basic interne

```
Function Minute(tmTime As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le nombre de minutes contenues l'heure exprimée par le paramètre *tmTime*.

Entrée

- *tmTime* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Dim MyStr
```

MyStr=Minute(AmDate())'Renvoie le nombre de minutes écoulées dans l'heure courante par exemple "45" s'il est actuellement 15:45:30

MkDir()

Syntaxe Basic interne

```
Function MkDir(strMkDirectory As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Crée un répertoire.

Entrée

- `strMkDirectory` : Chemin complet du répertoire à créer.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Month()

Syntaxe Basic interne

```
Function Month(tmDate As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou	X

	Utilisable
d'un lien	
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le mois contenu dans la date exprimée par le paramètre *tmDate*.

Entrée

- *tmDate* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Dim MyStr
MyStr=Month(AmDate())'Renvoie le mois courant
```

Name()

Syntaxe Basic interne

```
Function Name(strSource As String, strDest As String)
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renomme un fichier.

Entrée

- `strSource` : Chemin complet du fichier à renommer.
- `strDest` : Nouveau nom du fichier.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Now()

Syntaxe Basic interne

```
Function Now() As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la date et l'heure courantes.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

NPER()

Syntaxe Basic interne

```
Function NPER(dblRate As Double, dblPmt As Double, dblPV As Double,
dblFV As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le nombre d'échéances d'une annuité basée sur des versements constants et périodiques, et sur un taux d'intérêt fixe.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *dblPmt* : Ce paramètre indique le montant du paiement à effectuer à chaque échéance. Le paiement comporte généralement le principal et les intérêts.
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.
- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Les sommes versées (exprimées notamment par le paramètre *Pmt*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

Oct()

Syntaxe Basic interne

```
Function Oct(dValue As Double) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	

	Utilisable
APIs AssetCenter Web	X

Description

Renvoie la valeur octale d'un nombre.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître la valeur octale.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

ParseDMYDate()

Syntaxe Basic interne

```
Function ParseDMYDate(strDate As String) As Date
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie un objet Date (au sens Basic) à partir d'une date formatée comme suit :

`dd/mm/yyyy`

Entrée

- *`strDate`* : Date stockée sous la forme d'une chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

ParseMDYDate()

Syntaxe Basic interne

```
Function ParseMDYDate(strDate As String) As Date
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie un objet Date (au sens Basic) à partir d'une date formatée comme suit :

`mm/dd/yyyy`

Entrée

- *`strDate`* : Date stockée sous la forme d'une chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

ParseYMDDate()

Syntaxe Basic interne

```
Function ParseYMDDate(strDate As String) As Date
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie un objet Date (au sens Basic) à partir d'une date formatée comme suit :

```
yyyy/mm/dd
```

Entrée

- *strDate* : Date stockée sous la forme d'une chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

PMT()

Syntaxe Basic interne

```
Function PMT(dblRate As Double, iNper As Long, dblPV As Double, dblFV
As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou	X

	Utilisable
d'un lien	
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le montant d'une annuité basée sur des versements constants et périodiques, et sur un taux d'intérêt fixe.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *iNper* : Ce paramètre contient le nombre total d'échéances de l'opération financière.
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.
- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)

√ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les paramètres *`Rate`* et *`Nper`* doivent être calculés à l'aide d'échéances exprimées dans les mêmes unités.
- Les sommes versées (exprimées notamment par le paramètre *`Pmt`*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

PPMT()

Syntaxe Basic interne

```
Function PPMT(dblRate As Double, iPer As Long, iNper As Long, dblPV As
Double, dblFV As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le montant du remboursement du capital, pour une échéance donnée, d'une annuité basée sur des versements constants et périodiques et sur un taux d'intérêt fixe.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *iPer* : Ce paramètre indique la période concernée par le calcul, comprise entre 1 et la valeur du paramètre *Nper*.
- *iNper* : Ce paramètre contient le nombre total d'échéances de l'opération financière
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.

- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les paramètres *Rate* et *Nper* doivent être calculés à l'aide d'échéances exprimées dans les mêmes unités.
- Les sommes versées (exprimées notamment par le paramètre *Pmt*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

PV()

Syntaxe Basic interne

```
Function PV(dblRate As Double, iNper As Long, dblPmt As Double, dblFV
As Double, iType As Long) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le montant actuel d'une annuité basée sur des échéances futures constantes et périodiques, et sur un taux d'intérêt fixe.

Entrée

- *dblRate* : Ce paramètre indique le taux d'intérêt par échéance. Par exemple pour un prêt à taux d'intérêt annuel de 6%, remboursé suivant une périodicité mensuelle, le taux par échéance est de :

```
0,06/12=0,005 soit 0,5%
```

- *iNper* : Ce paramètre contient le nombre total d'échéances de l'opération financière.
- *dblPmt* : Ce paramètre indique le montant du paiement à effectuer à chaque échéance. Le paiement comporte généralement le principal et les intérêts.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.

- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les paramètres *Rate* et *Nper* doivent être calculés à l'aide d'échéances exprimées dans les mêmes unités.
- Les sommes versées (exprimées notamment par le paramètre *Pmt*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.

Randomize()

Syntaxe Basic interne

```
Function Randomize(lValue As Long)
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Initialise le générateur de nombres aléatoires.

Entrée

- `lValue` : Paramètre optionnel utilisé pour initialiser le générateur de nombres aléatoires de la fonction `Rnd` en lui donnant une nouvelle valeur initiale. Si ce paramètre est omis, la valeur renvoyée par l'horloge système est utilisée comme valeur initiale.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

RATE()

Syntaxe Basic interne

```
Function RATE(iNper As Long, dblPmt As Double, dblFV As Double, dblPV
As Double, iType As Long, dblGuess As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie le taux d'intérêt par échéance pour une annuité.

Entrée

- `iNper` : Ce paramètre contient le nombre total d'échéances de l'opération financière.

- *dblPmt* : Ce paramètre indique le montant du paiement à effectuer à chaque échéance. Le paiement comporte généralement le principal et les intérêts.
- *dblFV* : Ce paramètre contient la valeur future ou le solde que vous souhaitez obtenir après avoir effectué le dernier paiement. En règle générale, et dans le cas d'un remboursement d'un emprunt en particulier, ce paramètre prend la valeur "0". En effet, une fois toutes les échéances remboursées, la valeur de l'emprunt est nulle.
- *dblPV* : Ce paramètre contient la valeur actuelle (ou somme globale) d'une série de paiements devant être effectués dans le futur.
- *iType* : Ce paramètre indique la date d'échéances des paiements. Il peut prendre les valeurs suivantes :
  - √ *0* si les paiements sont dus à terme échu (c'est à dire en fin de période)
  - √ *1* si les paiements sont dus à terme à échoir (c'est à dire en début de période)
- *dblGuess* : Ce paramètre contient la valeur estimée du taux d'intérêt par échéance.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

- Les sommes versées (exprimées notamment par le paramètre *Pmt>*) sont représentées par des nombres négatifs. Les sommes reçues sont représentées par des nombres positifs.
- Cette fonction effectue les calculs par itération, en commençant par la valeur attribuée au paramètre *Guess*. Si aucun résultat n'est trouvé au bout de vingt itérations, la fonction échoue.

RemoveRows()

Syntaxe Basic interne

```
Function RemoveRows(strList As String, strRowNames As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Supprime dans une liste les lignes identifiées par le paramètre *strRowNames*.

Cette fonction est utile lors du traitement des valeurs d'un contrôle de type "ListBox". Les valeurs d'un tel contrôle sont représentées par des chaînes bi-dimensionnelles dont les caractéristiques sont les suivantes :

- Le caractère "|" est utilisé comme séparateur de colonnes.
- Le caractère "," est utilisé comme séparateur de lignes.

- Chaque ligne est terminée par un identifiant unique situé à droite du signe "="

Entrée

- *strList* : Chaîne source contenant les valeurs d'un contrôle "ListBox" à traiter.
- *strRowNames* : Identifiants des lignes à supprimer. Les identifiants sont séparés par des virgules.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr

MyStr=RemoveRows("a1|a2=a0,b1|b2=b0", "a0,c0") 'Renvoie "b1|b2=b0"
```

Replace()

Syntaxe Basic interne

```
Function Replace(strData As String, strOldPattern As String,
strNewPattern As String, bCaseSensitive As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Remplace toutes les occurrences du paramètre `strOldPattern` par la valeur du paramètre `strNewPattern` au sein de la chaîne de caractères contenue dans `strData`. La recherche de `strOldPattern` peut tenir compte ou non de la casse en fonction de la valeur du paramètre `bCaseSensitive`.

Entrée

- `strData` : Chaîne de caractères contenant les occurrences à remplacer.
- `strOldPattern` : Occurrence à recherche dans la chaîne de caractères contenue dans `strData`.
- `strNewPattern` : Texte remplaçant toute occurrence trouvée.
- `bCaseSensitive` : En fonction de la valeur de ce paramètre, la recherche respecte (=1) ou non (=0) la casse.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=Replace("toimoitoimoitoi", "toi", "moi",0) 'Renvoie "moimoimoimoimoi"

MyStr=Replace("toimoitoimoitoi", "Toi", "moi",1) 'Renvoie "toimoitoimoitoi"

MyStr=Replace("toimoiToimoitoi", "Toi", "moi",1) 'Renvoie "toimoimoimoitoi"

Right()

Syntaxe Basic interne

```
Function Right(strString As String, iNumber As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de	X

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie iNumber caractères d'une chaîne en partant de la droite.

Entrée

- *strString* : Chaîne de caractères à traiter.
- *iNumber* : Nombre de caractères à renvoyer.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Sub Main ()
   Dim LWord, Msg, RWord, SpcPos, UsrInp    ' Declare variables.
   Msg = "Enter two words separated by a space."
   UsrInp = InputBox(Msg)    ' Get user input.
   print UsrInp
   SpcPos = InStr(1, UsrInp, " ")    ' Find space.
```

```
    If SpcPos Then
       LWord = Left(UsrInp, SpcPos - 1)    ' Get left word.
       print "LWord: ";  LWord
       RWord = Right(UsrInp, Len(UsrInp) - SpcPos)    ' Get right word.
       Msg = "The first word you entered is " & LWord
       Msg = Msg & "." & " The second word is "
       Msg = "The first word you entered is <" & LWord & ">"
       Msg = Msg & RWord & "."
    Else
       Msg = "You didn't enter two words."
    End If
    Print Msg    ' Display message.
    MidTest = Mid("Mid Word Test", 4, 5)
    Print MidTest
End Sub
```

RightPart()

Syntaxe Basic interne

```
Function RightPart(strFrom As String, strSep As String, bCaseSensitive
As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait la portion d'une chaîne de caractères située à droite du séparateur précisé dans le paramètre `strSep`.

La recherche du séparateur s'effectue de la droite vers la gauche.

La recherche peut tenir compte ou non de la casse en fonction de la valeur du paramètre `bCaseSensitive`.

Entrée

- `strFrom` : Chaîne source à traiter.
- `strSep` : Caractère utilisé comme séparateur dans la chaîne source.
- `bCaseSensitive` : En fonction de la valeur de ce paramètre, la recherche respecte (=1) ou non (=0) la casse.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Ces exemples illustrent l'utilisation des fonctions `LeftPart`, `LeftPartFromRight`, `RightPart` et `RightPartFromLeft` sur une même chaîne de caractères: "Ceci_est_un_test" :

```
LeftPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci".

```
LeftPartFromRight("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci_est_un".

```
RightPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "test".

```
RightPartFromLeft("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "est_un_test".

RightPartFromLeft()

Syntaxe Basic interne

```
Function RightPartFromLeft(strFrom As String, strSep As String,
bCaseSensitive As Long) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Extrait la portion d'une chaîne de caractères située à droite du séparateur précisé dans le paramètre `strSep`.

La recherche du séparateur s'effectue de la gauche vers la droite.

La recherche peut tenir compte ou non de la casse en fonction de la valeur du paramètre `bCaseSensitive`.

Entrée

- `strFrom` : Chaîne source à traiter.
- `strSep` : Caractère utilisé comme séparateur dans la chaîne source.
- `bCaseSensitive` : En fonction de la valeur de ce paramètre, la recherche respecte (=1) ou non (=0) la casse.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Ces exemples illustrent l'utilisation des fonctions `LeftPart`, `LeftPartFromRight`, `RightPart` et `RightPartFromLeft` sur une même chaîne de caractères: "Ceci_est_un_test" :

```
LeftPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci".

```
LeftPartFromRight("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "Ceci_est_un".

```
RightPart("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "test".

```
RightPartFromLeft("Ceci_est_un_test","_",0)
```

Renvoie la chaîne "est_un_test".

RmDir()

Syntaxe Basic interne

```
Function RmDir(strRmDirectory As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	

	Utilisable
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Détruit un répertoire.

Entrée

- `strRmDirectory` : Chemin complet du répertoire à détruire.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Rnd()

Syntaxe Basic interne

```
Function Rnd(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie une valeur contenant un nombre aléatoire.

Entrée

- `dValue` : Paramètre optionnel dont la valeur définit le mode de génération adopté par la fonction :
  - √ Inférieur à zéro : Le même nombre est généré à chaque fois.
  - √ Supérieur à zéro : Nombre aléatoire suivant dans la série.
  - √ Egal à zéro : Dernier nombre aléatoire généré.
  - √ Omis : Nombre aléatoire suivant dans la série.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Avant d'appeler cette fonction, vous devez utiliser la fonction `Randomize`, sans aucun paramètre, pour initialiser le générateur de nombres aléatoires.

Exemple

```
Dim MyNumber

Randomize

MyNumber= Int((10*Rnd)+1) 'Renvoie une valeur aléatoire comprise entre 1 et 10.
```

RTrim()

Syntaxe Basic interne

```
Function RTrim(strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de	X

	Utilisable
caractéristique	
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Supprime tous les espaces suivant le dernier caractère (qui n'est pas un espace) d'une chaîne.

Entrée

- *`strString`* : Chaîne de caractères à traiter.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
' This example uses the LTrim and RTrim functions to strip
leading ' and trailing spaces, respectively, from a string
variable.

' It uses the Trim function alone to strip both types of spaces.

' LCase and UCase are also shown in this example as well as the use

' of nested function calls
```

```
Sub Main
   MyString = "  <-Trim->  "    ' Initialize string.
   TrimString = LTrim(MyString)    ' TrimString = "<-Trim->  ".
   Print "|" & TrimString & "|"
   TrimString = LCase(RTrim(MyString))    ' TrimString = "  <-trim->".
   Print "|" & TrimString & "|"
   TrimString = LTrim(RTrim(MyString))    ' TrimString = "<-Trim->".
   Print "|" & TrimString & "|"
   ' Using the Trim function alone achieves the same result.
   TrimString = UCase(Trim(MyString))    ' TrimString = "<-TRIM->".
   Print "|" & TrimString & "|"
End Sub
```

Second()

Syntaxe Basic interne

```
Function Second(tmTime As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le nombre de secondes contenu dans la l'heure exprimée par le paramètre *tmTime*.

Entrée

- *tmTime* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.

Exemple

```
Dim MyStr
```

MyStr=Second(AmDate())'Renvoie le nombre de secondes écoulées dans l'heure courante par exemple "30" s'il est actuellement 15:45:30

SetSubList()

Syntaxe Basic interne

```
Function SetSubList(strValues As String, strRows As String,
strRowFormat As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Définit les valeurs d'une sous-liste pour un contrôle "ListBox".

Entrée

- `strValues` : Chaîne source contenant les valeurs d'un contrôle "ListBox" à traiter.
- `strRows` : Liste de valeurs à ajouter ou à substituer à celles de la chaîne contenue dans le paramètre `strValues`. Les valeurs sont séparées par le caractère "|". Les lignes traitées sont identifiées par leur identifiant, situé à droite du signe "=". Les lignes inconnues de sont pas traitées.
- `strRowFormat` : Instructions de formatage de la sous-liste. Les instructions sont séparées par le caractère "|". Ce paramètre possède les caractéristiques suivantes :
  - √ "1" représente les informations contenues dans la première colonne de la sous-liste.
  - √ "i-j" peut être utilisé pour définir un ensemble de colonnes.

√ "-" prend en compte toutes les colonnes.

Une colonne inconnue ne renvoie aucune valeur.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "A2|A1=a0, B2|B1=b0", "2|1") 'Renvoie "A1|A2|a3=a0,B1|B2|b3=b0,c1|c2|c3=c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "Z2=*,B2=b0", "2") 'Renvoie "a1|Z2|a3=a0,b1|B2|b3=b0,c1|Z2|c3=c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "B5|B6|B7=b0,C5|C6,C7=c0", "5-7") 'Renvoie "a1|a2|a3=a0,b1|b2|b3|||B5|B6|B7=b0,c1|c2|c3|||C5|C6|C7=c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "B1|B2|B3|B4=b0", "-") 'Renvoie "a1|a2|a3=a0,B1|B2|B3|B4=b0,c1|c2|c3=c0"

MyStr=SubList("A|B|C,D|E|F", "X=*", "2") 'Renvoie "A|X|C,D|X|F"

Sgn()

Syntaxe Basic interne

```
Function Sgn(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie une valeur indiquant le signe d'un nombre.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître le signe.

Sortie

La fonction peut renvoyer une des valeurs suivante :

- 1 : Le nombre est supérieur à zéro.
- 0 : Le nombre est égal à zéro.
- -1 : Le nombre est inférieur à zéro.

Shell()

Syntaxe Basic interne

```
Function Shell(strExec As String) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Lance un programme exécutable.

Entrée

- `strExec` : Chemin complet de l'exécutable à lancer.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyId
```

MyId=Shell("C:\WinNT\notepad.exe")

Sin()

Syntaxe Basic interne

```
Function Sin(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X

	Utilisable
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le sinus d'un nombre, exprimé en radians.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître le sinus.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Space()

Syntaxe Basic interne

```
Function Space(iSpace As Long) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Crée une chaînes de caractères comprenant le nombre d'espaces indiqué par le paramètre `iSpace`.

Entrée

- `iSpace` : Nombre d'espaces à insérer dans la chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Remarques

Cette fonction peut servir à formater des chaînes ou à effacer des données dans des chaînes de longueur fixe.

Exemple

```
Dim MyString

' Renvoie une chaîne de 10 espaces.

MyString = Space(10)

' Insère 10 espaces entre deux chaînes.

MyString = "Espace" & Space(10) & "inséré"
```

Sqr()

Syntaxe Basic interne

```
Function Sqr(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X

	Utilisable
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la racine carrée d'un nombre.

Entrée

- *dValue* : Nombre dont vous souhaitez connaître la racine carrée.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Str()

Syntaxe Basic interne

```
Function Str(strValue As String) As String
```

Champ d'application

Version : 3.00

	Utilisable

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit un nombre en une chaîne de caractères.

Entrée

- `strValue` : nombre à convertir en chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

StrComp()

Syntaxe Basic interne

```
Function StrComp(strString1 As String, strString2 As String,
iOptionCompare As Long) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Effectue la comparaison entre deux chaînes de caractères.

Entrée

- *strString1* : Première chaîne de caractères.
- *strString2* : Deuxième chaîne de caractères.

- *iOptionCompare* : type de comparaison. Ce paramètre peut prendre la valeur "0" pour une comparaison binaire, ou "1" pour une comparaison du texte des deux chaînes.

Sortie

- -1 : *strString1* est supérieure à *strString2*.
- 0 : *strString1* est égale à *strString2*.
- 1 : *strString1* est inférieure à *strString2*.

String()

Syntaxe Basic interne

```
Function String(iCount As Long, strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie une chaîne composée de *iCount* fois le caractère *strString*.

Entrée

- *iCount* : Nombre d'occurrences du caractère *strString*.
- *strString* : caractère utilisé pour la composition de la chaîne.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

SubList()

Syntaxe Basic interne

```
Function SubList(strValues As String, strRows As String, strRowFormat
As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X

	Utilisable
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie une sous-liste d'une liste de valeurs contenue dans une chaîne de caractères représentant les valeurs d'un contrôle "ListBox".

Entrée

- `strValues` : Chaîne source contenant les valeurs d'un contrôle "ListBox" à traiter.
- `strRows` : Identifiants des lignes à inclure dans la sous-liste. Les identifiants sont séparés par une virgule. Certains jokers sont acceptés :
  - √ "*" inclut tous les identifiants dans la sous-liste.
  - √ Un identifiant inconnu renvoie une valeur vide pour la sous-liste.
- `strRowFormat` : Instructions de formatage de la sous-liste. Les instructions sont séparées par le caractère "|". Ce paramètre possède les caractéristiques suivantes :
  - √ "1" représente les informations contenues dans la première colonne de la liste dont on extrait une sous-liste.
  - √ "0" représente l'identifiant de la ligne de la liste dont on extrait une sous-liste.
  - √ "*" représente les informations contenues dans toutes les colonnes (à l'exception de l'identifiant de la ligne).
  - √ Une colonne inconnue ne renvoie aucune valeur.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "a0,b0,a0", "3|2|3") 'Renvoie "a3|a2|a3,b3|b2|b3,a3|a2|a3"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "*|0") 'Renvoie "a1|a2|a3|a0,b1|b2|b3|b0,c1|c2|c3|c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "*=0") 'Renvoie "a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "999=0") 'Renvoie "=a0,=b0,=c0"

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "z0", "*=0") 'Renvoie ""

MyStr=SubList("a1|a2|a3=a0,b1|b2|b3=b0,c1|c2|c3=c0", "*", "=1") 'Renvoie "=a1,=b1,=c1"

MyStr=SubList("A|B|C,D|E|F", "*", "2=0") 'Renvoie "B,E"

Tan()

Syntaxe Basic interne

```
Function Tan(dValue As Double) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie la tangente d'un nombre, exprimé en radians.

Entrée

- `dValue` : Nombre dont vous souhaitez connaître la tangente.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Time()

Syntaxe Basic interne

```
Function Time() As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'heure courante.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.

- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Timer()

Syntaxe Basic interne

```
Function Timer() As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le nombre de secondes écoulées depuis 12:00 AM.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

TimeSerial()

Syntaxe Basic interne

```
Function TimeSerial(iHour As Long, iMinute As Long, iSecond As Long) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie une heure formatée en fonction des paramètres *iHour*, *iMinute* et *iSecond*.

Entrée

- *iHour* : Heure.
- *iMinute* : Minutes.
- *iSecond* : Secondes.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

Chacun de ces paramètres peut prendre pour valeur une expression numérique représentant un nombre d'heures, de minutes ou de secondes. Ainsi l'exemple suivant :

```
TimeSerial(12-8, -10, 0)
```

renvoie la valeur :

```
3:50:00
```

Lorsque la valeur d'un paramètre est en dehors de l'intervalle de valeurs généralement admis (c'est à dire 0-59 pour les minutes et les secondes et 0-23 pour les heures), elle est convertie vers le paramètre immédiatement supérieur. Ainsi, si vous entrez "75" comme valeur pour le paramètre *iMinute*, ce dernier sera interprété comme 1 heure et 15 minutes.

L'exemple suivant :

```
TimeSerial (16, 50, 45)
```

renvoie la valeur :

```
16:50:45
```

TimeValue()

Syntaxe Basic interne

```
Function TimeValue(tmTime As Date) As Date
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Cette fonction renvoie la partie heure d'une valeur "Date+Heure"

Entrée

- *tmTime* : Date au format "Date+Heure".

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

L'exemple suivant :

```
TimeValue ("1999/09/24 15:00:00")
```

renvoie la valeur :

```
15:00:00
```

Trim()

Syntaxe Basic interne

```
Function Trim(strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Supprime tous les espaces précédant le premier caractère (qui n'est pas un espace) d'une chaîne et tous les espaces suivant le dernier caractère (qui n'est pas un espace) d'une chaîne.

Entrée

- *`strString`* : Chaîne de caractères à traiter.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
' This example uses the LTrim and RTrim functions to strip
leading ' and trailing spaces, respectively, from a string
variable.

' It uses the Trim function alone to strip both types of spaces.

' LCase and UCase are also shown in this example as well as the use

' of nested function calls
```

```
Sub Main
   MyString = "  <-Trim->  "    ' Initialize string.
   TrimString = LTrim(MyString)    ' TrimString = "<-Trim->  ".
   Print "|" & TrimString & "|"
   TrimString = LCase(RTrim(MyString))    ' TrimString = "  <-trim->".
   Print "|" & TrimString & "|"
   TrimString = LTrim(RTrim(MyString))    ' TrimString = "<-Trim->".
   Print "|" & TrimString & "|"
   ' Using the Trim function alone achieves the same result.
   TrimString = UCase(Trim(MyString))    ' TrimString = "<-TRIM->".
   Print "|" & TrimString & "|"
End Sub
```

UCase()

Syntaxe Basic interne

```
Function UCase(strString As String) As String
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Passe tous les caractères d'une chaîne en majuscules.

Entrée

- *`strString`* : Chaîne de caractères à passer en majuscules.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
' This example uses the LTrim and RTrim functions to strip
leading ' and trailing spaces, respectively, from a string
variable.

' It uses the Trim function alone to strip both types of spaces.

' LCase and UCase are also shown in this example as well as the use

' of nested function calls


Sub Main
```

```
    MyString = "  <-Trim->  "    ' Initialize string.
    TrimString = LTrim(MyString)    ' TrimString = "<-Trim->  ".
    Print "|" & TrimString & "|"
    TrimString = LCase(RTrim(MyString))    ' TrimString = "  <-trim->".
    Print "|" & TrimString & "|"
    TrimString = LTrim(RTrim(MyString))    ' TrimString = "<-Trim->".
    Print "|" & TrimString & "|"
    ' Using the Trim function alone achieves the same result.
    TrimString = UCase(Trim(MyString))    ' TrimString = "<-TRIM->".
    Print "|" & TrimString & "|"
End Sub
```

UnEscapeSeparators()

Syntaxe Basic interne

```
Function UnEscapeSeparators(strSource As String, strEscChar As String)
As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X

	Utilisable
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Supprime tous les caractères d'échappement d'une chaîne de caractères.

Entrée

- *strSource* : Chaîne de caractères à traiter.
- *strEscChar* : Caractère d'échappement à supprimer.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr

MyStr=UnEscapeSeparators("toi\|moi\|toi\|", "\") 'Renvoie la valeur "toi|moi|toi|"
```

Union()

Syntaxe Basic interne

```
Function Union(strListOne As String, strListTwo As String, strSeparator
As String, strEscChar As String) As String
```

Champ d'application

Version : 3.5

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Rassemble deux chaînes de caractères délimitées par des séparateurs. Les doublons sont supprimés.

Entrée

- `strListOne` : Première chaîne de caractères.
- `strListTwo` : Deuxième chaîne de caractères.

- *strSeparator* : Séparateur utilisé pour délimiter les éléments contenus dans les chaînes.
- *strEscChar* : Caractère d'échappement. Si ce caractère préfixe le séparateur, ce dernier est ignoré.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

Exemple

```
Dim MyStr
```

MyStr=Union("a1|a2,b1|b2", "a1|a3,b1|b2", ",", "\") 'Renvoie la valeur "a1|a2,b1|b2,a1|a3"

MyStr=Union("a1|a2,b1|b2", "a1|a3\,b1|b2", ",", "\") 'Renvoie la valeur "a1|a2,b1|b2,a1|a3\,b1|b2"

Val()

Syntaxe Basic interne

```
Function Val(strString As String) As Double
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X

	Utilisable
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Convertit une chaîne de caractères représentant un nombre en un nombre de type "Double".

Entrée

- `strString` : Chaîne à convertir.

Sortie

En cas d'erreur, deux cas de figure se présentent :

- Sous AssetCenter, l'exécution du script contenant la fonction est interrompue et un message d'erreur est envoyé à l'utilisateur.
- Dans le cas d'un appel par le biais d'un programme externe, vous devez appeler la fonction `AmLastError` (et éventuellement la fonction `AmLastErrorMsg`) pour savoir si une erreur s'est produite (et son message associé).

WeekDay()

Syntaxe Basic interne

```
Function WeekDay(tmDate As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie le jour de la semaine contenu dans la date exprimée par le paramètre *tmDate*.

Entrée

- *tmDate* : Paramètre au format Date+Heure à traiter.

Sortie

Le nombre retourné correspond à un jour de la semaine, le "1" représentant le dimanche, le "2" le lundi, ..., le "7" le samedi.

Year()

Syntaxe Basic interne

```
Function Year(tmDate As Date) As Long
```

Champ d'application

Version : 3.00

	Utilisable
Fonction Builtin	X
AssetCenter APIs	
Script de configuration d'un champ ou d'un lien	X
Script de définition d'un paramètre de caractéristique	X
Action de type "Script"	X
Script d'un assistant	X
Script FINISH.DO d'un assistant	
APIs AssetCenter Web	X

Description

Renvoie l'année contenue dans la date exprimée par le paramètre *tmDate*.

Entrée

- *tmDate* : Paramètre au format Date+Heure à traiter.

Sortie

- 0 : La fonction s'est exécutée normalement.
- Non nul : Code d'erreur.


Peregrine
SYSTEMS®
The Infrastructure Management Company™