

BIREME / PAHO / WHO

Centre d'Information sur les Sciences de la Santé pour l'Amérique Latine et les Caraïbes

Langage de formatage CISIS

Version 4.x

Sao Paulo - 2007

Copyright © 2007 - BIREME / PAHO / WHO

Langage de formatage CISIS

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.2 ou ultérieure publiée par la Free Software Foundation; sans les Sections Inaltérables, sans les les textes de première page de couverture ou les textes de dernière page de couverture. Une copie de cette Licence est incluse dans la section appelée "GNU Free Documentation License".

Notice de catalogage

BIREME / PAHO / WHO (Brésil)

Langage de formatage CISIS. / BIREME / PAHO / WHO.

Sao Paulo : BIREME / PAHO / WHO, 2007.

43 p.

1. Manuel utilisateur. 2. Accès à l'information. 3. Systèmes d'information. 4. Gestion de l'information. 5. Santé publique. 6. Services de santé. I. BIREME II. Titre

Attention - Aucune mention faite dans ce document aux organismes, institutions, personnes ou produits, ne peut être considérée comme une approbation ou une recommandation donnée par BIREME / PAHO / WHO, et ne signifie pas qu'y ait préférence à d'autres de nature semblable, cités ou non.

BIREME / PAHO / WHO

Centre d'Information sur les Sciences de la Santé pour l'Amérique Latine et les Caraïbes

Rua Botucatu 862 V Clementino

Ce document a été produit avec le "Documents Conformation Methodology (NorDoc)" développé par BIREME.

Table des matières

Abréviations utilisées	IV
Comment utiliser ce manuel	VI
Préface	1
A propos de BIREME	1
La Bibliothèque virtuelle sur la santé (BVS)	3
Structure de la liste de référence	5
Commandes et fonctions	6
# % " "" () / /* @	6
A	9
B	10
C	10
D	12
E	13
F	14
G	15
I	15
L	16
M	18
N	20
P	22
R	23
S	26
T	29
V	30
W	32
X	33
Références bibliographiques	34
Glossaire	35

Abréviations utilisées

- ANSI. American National Standards Institute [Institut national américain de normalisation].
- ASCII. American Standard Code for Information Interchange [Code standard américain pour l'échange d'information].
- BIREME. Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud [Centre d'Information sur les Sciences de la Santé pour l'Amérique Latine et les Caraïbes].
- BVS. Biblioteca Virtual en Salud [Bibliothèque virtuelle sur la santé].
- CDS. Computerized Documentation System [Système de documentation informatisée].
- FST. Field Selection Table [Table de sélection des champs].
- FTP. File Transfer Protocol [Protocole de transfert de fichiers].
- IFP. Inverted File Pointer [pointeur du fichier inversé].

- **ISIS.** Integrated Set of Information Systems [Ensemble intégré de systèmes d'information].
- **ISO.** International Organization for Standardization [Organisation internationale de normalisation].
- **LILACS.** Literatura Latinoamericana y del Caribe en Ciencias de la Salud [Littérature Latino-Américaine et des Caraïbes en sciences de la santé].
- **OMS.** Organización Mundial de la Salud [Organisation mondiale de la santé].
- **OPS.** Organización Panamericana de la Salud [Organisation panaméricaine de la santé].
- **UNESCO.** United Nations Educational, Scientific and Cultural Organization [Organisation des Nations Unies pour l'éducation, la science et la culture].

Comment utiliser ce manuel

Ce manuel est classé par ordre alphabétique de commandes ou de fonctions. Sont également énumérés par ordre alphabétique les noms attribués à chaque commande ou fonction pour aider l'utilisateur à retrouver les termes souhaités.

Pour chaque commande ou fonction, toutes les informations sur son utilisation sont décrites sous forme de tableau, en donnant la syntaxe et des exemples.

En complément du document, un glossaire, une liste d'abréviations et des références bibliographiques d'autres documents intéressants.

Préface

A propos de BIREME

Année après année, BIREME a rempli sa mission, à savoir être un centre spécialisé dans l'information scientifique et technique dans le domaine de la santé pour la région de l'Amérique latine et des Caraïbes. Fondée au Brésil en 1967, sous le nom de Bibliothèque régionale de médecine (d'où l'acronyme BIREME), elle a toujours répondu à la demande croissante en littérature scientifique à jour de la part des systèmes brésiliens de santé, des communautés de chercheurs, des professionnels et des étudiants du secteur de la santé. Puis en 1982, son nom changea en Centre d'Information sur les Sciences de la Santé pour l'Amérique Latine et les Caraïbes pour mieux exprimer ses fonctions orientées vers le renforcement et l'extension du flux d'information scientifique et technique sur la santé dans toute la région, mais elle a conservé son sigle.

Le travail en réseau, basé sur la décentralisation, le développement des capacités locales, le partage des ressources de l'information, le développement de produits et de services coopératifs, la conception des méthodologies communes, a toujours été le fondement du travail de coopération technique de BIREME. C'est de cette manière que le centre est devenu un modèle international qui promeut la qualification des professionnels de l'information au niveau managérial et

technique afin que ceux-ci adoptent les paradigmes de l'information et de la communication qui répondent le mieux aux besoins locaux.

Les principes les plus importants qui sont à l'origine du BIREME et qui sont les fondements de son existence sont :

l'accès à l'information scientifique et technique sur la santé est essentiel pour le développement de la santé ;

le besoin de développer la capacité des pays de l'Amérique latine et des Caraïbes à exploiter leurs sources d'information scientifique et technique sur la santé de manière coopérative et efficace ;

la nécessité de promouvoir l'utilisation et de répondre aux demandes d'informations scientifique et technique sur la santé venant des gouvernements, des systèmes de santé, des établissements d'enseignement et de recherche.

BIREME, en tant que centre spécialisé de l'Organisation panaméricaine de la santé (OPS)/ Organisation mondiale de la santé (OMS), coordonne et réalise des activités de coopération technique en gestion de l'information et des connaissances scientifiques dans le but de renforcer et d'augmenter le flux d'information scientifique sur la santé au Brésil, dans les autres pays de l'Amérique latine et aux Caraïbes, comme condition principale pour le développement de la santé, y compris sa planification, gestion, promotion, recherche et éducation.

La convention qui soutient BIREME est renouvelée tous les cinq ans par les membres du Comité consultatif national de l'établissement (PAHO, ministère de santé brésilien, ministère brésilien de l'éducation et de la culture, secrétariat d'Etat à la santé de l'état de São Paulo, et université fédérale de São Paulo - Unifesp). Le dernier fournit l'infrastructure physique nécessaire pour l'établissement de l'institution.

En 2004 BIREME a pris la responsabilité de devenir un établissement basé sur la connaissance.

La Bibliothèque virtuelle sur la santé (BVS)

Avec l'apparition et le développement d'Internet comme moyen prédominant d'information et de communication, le modèle de coopération technique de BIREME a évolué depuis 1998 vers la construction et le développement de la Bibliothèque virtuelle sur la santé (BVS) comme espace commun pour la convergence du travail coopératif des producteurs, intermédiaires et utilisateurs d'information. La BVS promeut le développement d'un réseau de sources d'information scientifique et technique avec l'accès universel à Internet. Pour la première fois il existe une réelle possibilité d'accès équitable à l'information sur la santé.

Pour BIREME, la Bibliothèque virtuelle sur la santé est un modèle pour la gestion de l'information et de la connaissance qui inclut la coopération et la convergence entre les établissements, les systèmes, les réseaux et les initiatives des producteurs, des intermédiaires et des utilisateurs dans l'exploitation des réseaux des sources d'information locales, nationales, régionales et internationales, en privilégiant l'accès libre et universel.

Aujourd'hui, tous les pays d'Amérique latine et des Caraïbes (Région) participent directement ou indirectement aux produits et services coopératifs offerts par la BVS, qui inclut plus de 1.000 établissements répartis dans plus de 30 pays.

La BVS est simulée dans un espace virtuel d'Internet formé par une collection ou un réseau de sources d'information sur la santé dans la Région. Les utilisateurs de niveaux et de lieux différents peuvent intervenir et naviguer dans l'espace d'une ou plusieurs sources d'information, indépendamment de l'endroit où ils se trouvent. Les sources d'information sont produites, mises à jour, stockées et exploitées sur Internet par des producteurs, des intégrateurs et des intermédiaires, de façon décentralisée, en suivant les méthodologies communes pour leur intégration dans la BVS.

La BVS organise l'information dans une structure qui intègre et relie des bases de données de référence, des annuaires spécialisés, des événements et des établissements, un catalogue des ressources de l'information disponibles sur Internet, des collections de textes intégraux avec entre autres la SciELO

(Bibliothèque électronique scientifique en ligne) collection de périodiques scientifiques, des services sélectifs de diffusion de l'information, des sources d'information pour soutenir l'éducation et la prise de décision, des nouvelles, des listes de discussion et une aide aux communautés virtuelles. L'espace de la BVS est donc un réseau dynamique et décentralisé des sources d'information à partir duquel il est possible de récupérer et extraire des informations pour soutenir les processus décisionnels dans le secteur de la santé.

La Bibliothèque virtuelle sur la santé peut être perçue comme une base partagée de la connaissance scientifique et technique dans le domaine de la santé qui est enregistrée, organisée et stockée dans le format électronique des pays de la Région, accessible de manière universelle sur Internet et compatibles avec les bases de données internationales.

Structure de la liste de référence

<spécification de format>	<nom>
Support :	<i>Standard</i> signifie que la commande/fonction a la même utilisation/résultat avec ISIS ou CISIS. Si <i>CISIS</i> est spécifié, signifie que la commande/fonction est uniquement disponible pour CISIS. Les commandes/fonctions qui ont été améliorées pour CISIS, sont mentionnées avec la notation <i>Standard/CISIS</i> . (rubrique toujours présente)
Type de fonction :	Spécifie le type de la valeur de retour de la fonction. Les valeurs possibles sont : booléen, chaîne de caractères et numérique. (rubrique uniquement applicable aux fonctions)
Syntaxe :	Notation formelle de l'utilisation d'une commande/fonction. (rubrique toujours présente)
Définition :	Explique l'utilisation de la commande/fonction.
Paramètres :	Explique les attributs supplémentaires de la commande/fonction.
Notes :	Explique les particularités, restrictions et différences entre ISIS et CISIS.
Exemples :	1 Donne un ou plusieurs exemples d'utilisation de la commande/fonction.
Voir aussi :	Liste des commandes et fonctions associées.

Commandes et fonctions

% " "" | | () /* @

retour à la ligne inconditionnel

Support :	Standard
Syntaxe :	#
Définition :	Commence un nouvelle ligne de manière inconditionnelle.
Exemple :	1 #,("adresse :" $v3(9,9)+ $; #), 2 (Nom : $ v1^n,c20$, Prénom : $ v1^s/$),###,
Voir aussi :	/ commande % commande

% **suppression de lignes vides**

Support :	Standard
Syntaxe :	%
Définition :	Supprime les lignes vides précédentes, si nécessaire.
Exemple :	1 Nom : $ v1^n,c20$, Prénom : $ v1^s,###,%,/$, 2 $v10/#,v20/#,v30/#,##,$
Voir aussi :	/ commande # commande

"chaîne" **littéral conditionnel**

Support :	Standard
-----------	----------

"chaîne" **littéral conditionnel**

Syntaxe :	"<texte>"<sélecteur de champ>"<texte>" "<texte>"<sélecteur de champ fictif (présent)> "<texte>"<sélecteur de champ fictif (absent)>
Définition :	Affiche le texte écrit entre guillemets seulement si <sélecteur de champ>, <sélecteur de champ fictif (présent)> ou <sélecteur de champ fictif (absent)>, retourne la valeur VRAI. Les littéraux préfixés et suffixés peuvent être utilisés ensemble, ils ne sont affichés que si le champ associé est présent. En cas d'association avec <sélecteur de champ fictif (présent)>, la sortie est générée seulement si le champ est présent. En cas d'association avec <sélecteur de champ fictif (absent)>, la sortie est générée seulement si le champ est absent.
Notes :	<texte> ne sera affiché qu'une seule fois, même si le champ est répétitif.
Exemples :	1 "Auteur : "v1^a, 2 "ce texte s'affiche si le champ 10 est présent"d10, 3 "ce texte s'affiche si le champ 10 est présent"n5,
Voir aussi :	'chaîne' commande chaîne commande d commande n commande v commande

'chaîne' **littéral inconditionnel**

Support :	Standard
Syntaxe :	'<texte>'
Définition :	Affiche de manière inconditionnelle le texte écrit entre apostrophes.
Notes :	Les littéraux inconditionnels peuvent être placés n'importe où dans un format d'affichage et peuvent être utilisés comme arguments de fonctions.
Exemples :	1 'ce texte sera toujours affiché', 2 'Nom : ',v1/,
Voir aussi :	"chaîne" commande chaîne commande

|chaîne| **littéral conditionnel répétitif**

Support :	Standard
Syntaxe :	<texte> <+><sélecteur de champ><+> <texte>

 chaîne 	littéral conditionnel répétitif
Définition :	Affiche le texte écrit entre barres verticales pour chaque occurrence d'un champ répétitif, seulement si le sélecteur de champ retourne la valeur VRAI. Associé à un champ répétitif, le fonctionnement de la commande peut être amélioré en utilisant l'opérateur <+>. Quand <+> est présent, le premier littéral préfixé et/ou le dernier littéral suffixé ne peut être affiché.
Notes :	Les littéraux préfixés et suffixés peuvent être utilisés ensemble, même avec l'opérateur <+>. Si un littéral préfixé ou suffixé est utilisé avec <+ > à l'extérieur d'un groupe répétitif, le contenu du littéral peut ne pas être affiché de manière satisfaisante. Si le champ n'est pas répétitif, le littéral n'est affiché qu'une seule fois.
Exemples :	1 (; +v1^s, , v1^n*0.1 .), 2 (v10 : , ,v11, - v12),
Voir aussi :	" chaîne " commande ' chaîne ' commande (format) commande v commande
(format)	groupe répétitif
Support :	Standard
Syntaxe :	(<format>)
Définition :	Applique le format écrit entre parenthèses à toutes les occurrences du groupe répétitif, séparément, ou seulement une fois si le champ n'est pas répétitif.
Notes :	Les groupes répétitifs ne peuvent pas être imbriqués.
Exemples :	1 (; +v2^s/), 2 (v1,c15,v2,c35,v3/), 3 (if iocc<=3 then f(iocc,1,0), - v3/ else '-> plus que 3'/, fi),
Voir aussi :	chaîne commande v commande
/	retour à la ligne conditionnel
Support :	Standard
Syntaxe :	/
Définition :	Commence une nouvelle ligne, sauf si la commande existe déjà en début de ligne.
Notes :	Plusieurs commandes de retour à la ligne conditionnel (/,/./,/) produisent le même résultat qu'une seule.
Exemples :	1 v1/, 2 v1/,v3/,v10/,mf/, 3 s(v1,v3,v10)/,

/	retour à la ligne conditionnel
---	---------------------------------------

Voir aussi : # commande
% commande

/*chaîne*/	commentaire
-------------------	--------------------

Support : CISIS
 Syntaxe : /* <commentaire> */
 Définition : Encadre une commentaire.
 Notes : Les commentaires peuvent être à cheval sur plusieurs lignes
 Exemples :
 1 /* ceci est un commentaire sur une seule ligne*/,
 2 /* ce commentaire commence ici
 et se termine ici */,
 3 if a(v10) /*and p(v20) */ then v20/ fi,

@	appel d'un format externe
---	----------------------------------

Support : CISIS
 Syntaxe : @<nom de fichier>
 Définition : Insère dans le format courant un format d'affichage complémentaire.
 Notes : <nomdefichier> peut donner le chemin d'accès complet au répertoire dans lequel est stocké le format externe. La syntaxe des commandes contenues dans le format externe est évaluée lorsque le format courant est exécuté. Il est indispensable d'encadrer <nomdefichier> par des virgules (,@<nom de fichier>).
 Exemples :
 1 @test.pft,v20,
 2 s(@c:\temp\test.pft,v3),
 3 if v1='L' then @large.pft, fi,

A

a(sélecteur de champ)	vérification de l'absence du champ
------------------------------	---

Support : Standard
 Type de fonction : Booléen
 Syntaxe : a(<sélecteur de champ>)
 Définition : Retourne la valeur VRAI si le champ est absent, ou FAUX dans le cas contraire.
 Notes : Il est possible d'utiliser tous les paramètres s'appliquant au sélecteur de champ, sauf la commande d'indentation.
 Exemples :
 1 if a(v12) then v13 else v12, fi,

a(sélecteur de champ) vérification de l'absence du champ

2 if **a**(v20^b) and p(v30) then v40/, fi,

Voir aussi :

p fonction
v commande

actualisation du champ

Voir :

proc fonction

affichage du contenu d'un fichier

Voir :

cat fonction

appel d'un format externe

Voir :

@ commande

B**break saut / sortie conditionnel(le)**

Support :

CISIS

Syntaxe :

break

Définition :

Interrompt l'exécution du format d'un groupe répétitif. A l'extérieur d'un groupe répétitif, l'exécution du format courant s'arrête.

Notes :

L'exécution reprend après la fin du groupe répétitif. Lors d'une utilisation dans une **fonction ref**, l'exécution continue après la fonction.

Exemples :

1 (if iocc > 10 then '10+ occurrences'/,**break** else v5^n|
|,v5^s/, fi,)

Voir aussi :

(format) commande

C**c colonne**

Support :

Standard

Syntaxe :

c<entier>

Définition :

Se déplace à la colonne spécifiée, dans la ligne courante ou la ligne suivante.

Exemples :

1 'Nom : ',c10,v1^n/,

c	colonne
----------	----------------

Voir aussi : 2 if p(v1^s) then c10,v1^s/, fi,
X commande

cat(fichier)	affichage du contenu d'un fichier
---------------------	--

Support : CISIS
Type de fonction : Chaîne de caractères
Syntaxe : **cat(<format>)**
Définition : Montre le contenu d'un fichier dont le nom est indiqué par
 <format>.
Exemples : 1 mfn,cat('fichier.html'),
 2 cat('document en cours'/, ,if v10='c' then 'fichier.txt' else
 'default.doc' fi),
 3 cat(v101),
Voir aussi : **s** fonction

commentaire	
--------------------	--

Voir : /***chaîne***/ commande

concaténation de chaîne de caractères	
--	--

Voir : **s** fonction

continue	saut conditionnel répétitif
-----------------	------------------------------------

Support : CISIS
Syntaxe : **continue**
Définition : Exécute la prochaine occurrence d'un champ répétitif s'il existe
 au moins un champ avec une telle occurrence.
Notes :
Exemples : 1 (if iocc = 1 then continue else v10/ fi),
 2 (f(iocc,1,0),'=',v70,continue/),
Voir aussi : **(format)** commande

conversion d'une chaîne de caractères en valeur numérique	
--	--

Voir : **val** fonction

conversion d'une valeur numérique en chaîne de caractères	
--	--

Voir : **f** fonction

D

d	sélecteur de champ fictif (présent)
Support :	Standard
Syntaxe :	d <étiquette de champ><sous-champ>
Définition :	Affiche le littéral conditionnel préfixé si le champ ou le sous-champ associé existe. On l'utilise en l'associant à un littéral conditionnel.
Notes :	Le sélecteur de champ fictif ne renvoie pas de valeur. Bug connu : dans un sous-champ répétitif, <sous-champ> est évalué seulement pour la première occurrence du champ.
Exemples :	1 "ce texte s'affiche si le champ 10 est présent" d 10, 2 "Nom : "v20(5,5)/, "Nom : "n20,v21(5,5)/,
Voir aussi :	"chaîne" commande n commande v commande
date date(mot clé)	date courante
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	date date (<mot clé>)
Définition :	Affiche la date courante du système. Dans le cas d'une utilisation sans paramètres, la date est affichée sous la forme : aaaammjj hhhmss s nnn où : aaaa = année mm = mois jj = jour hh = heure mm = minute ss = seconde s = jour de la semaine (0-6) nnn = nombre de jours écoulés depuis le 1 ^{er} janvier.
Paramètres :	Mots-clés DATETIME et DATEONLY .
Notes :	DATETIME affiche la date au format européen avec l'heure courante (jj/mm/aa hh:mm:ss), tandis que DATEONLY affiche la date au format européen sans l'heure courante.
Exemples :	1 'Nous sommes le ', date , 2 'Date courante : ', date (DATEONLY)/, 'Temps courant : ',mid(date (DATETIME),10,8)/,

date courante

Voir : **date** fonction

datex	génération d'une date
--------------	------------------------------

Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	datex (<format>)
Définition :	Génère une date au format DATE, équivalente au nombre de secondes écoulées depuis le 1 ^{er} janvier 1970 00:00:00, produite par <format>.
Notes :	Même limites que pour seconds()
Exemple :	mx null "pft= datex(1147780749) /"
Voir aussi :	date commande seconds fonction

définition de la largeur de ligne
--

Voir : **lw** fonction

définition d'une variable d'environnement
--

Voir : **putenv** fonction

E

e0 .. e9	définition de variables numériques
-----------------	---

Support :	CISIS
Type de fonction :	Numérique
Syntaxe :	e <n>:=<expression numérique>
Définition :	CISIS définit 10 variables numériques, de e0 à e9. Les variables sont initialisées à la valeur 0 chaque fois qu'un format est exécuté.
Notes :	Il est possible de changer la valeur de la variable pendant l'exécution du format. Une variable numérique peut être utilisée dans n'importe quelle partie du format qui requiert une valeur, par exemple comme opérande dans une expression du type if e1 +10<25 then ... fi Comme n'importe quelle valeur numérique, une variable numérique ne peut être utilisée directement mais doit d'abord être convertie en passant par la fonction f .
Exemples :	e1 :=val(v10 + 5)
Voir aussi :	s0 ... s9 commande f fonction

exécution conditionnelle en bloc

Voir : **select ... case ... elsecase ... endsel** commande

G

getenv(expression) valeur d'une variable d'environnement

Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	getenv (<format>)
Définition :	Retourne la valeur d'une variable d'environnement.
Notes :	Si <format> ne génère pas le nom d'une variable d'environnement valide, cela ne retourne aucune valeur.
Exemples :	<ol style="list-style-type: none"> 1 'Chemin courant : ',getenv('PATH'), 2 (v1 = ,getenv(v1)/),
Voir aussi :	putenv fonction

groupe répétitif

Voir : (format) commande

I

if ... then ... else ... fi instruction conditionnelle

Support :	Standard
Syntaxe :	if <expression booléenne> then <format-1> [else <format-2>] fi
Définition :	Exécute un bloc de spécifications de formatage (<format-1>) si le résultat du test de <expression booléenne> retourne la valeur VRAI. Exécute un autre bloc de spécifications de formatage (<format-2>) en utilisant la clause else uniquement si le résultat de <expression booléenne> retourne la valeur FAUX.
Notes :	La clause then précède le premier bloc. else est optionnel et, s'il est présent, doit être suivi d'un autre bloc. La clause fi termine toujours la commande et, si elle est absente, cela provoque une erreur de syntaxe. La commande if ... fi peut être à cheval sur plusieurs lignes, dans ce cas l'utilisation de l'indentation est recommandé.
Exemples :	<ol style="list-style-type: none"> 1 ,if instr(v5,'ab')>0 then ,v5/, fi, 2 ,if p(v10) then , Titre : v3, else , Autre titre : v4, fi,

indice d'occurrence

Voir : **iocc** fonction

insertion d'espaces

Voir : **x** commande

instr(chaine1,chaîne2)	recherche de chaîne de caractères
-------------------------------	--

Support :	CISIS
Type de fonction :	Numérique
Syntaxe :	instr (<format-1>,<format-2>)
Définition :	Retourne un nombre spécifiant la position de départ de la chaîne générée par <format-2>, dans la chaîne générée par <format-1>. Si la chaîne recherchée (<format-2>) ne se trouve pas dans <format-1> la fonction retourne 0.
Notes :	<format-1> et <format-2> doivent générer des chaînes de caractères, dans le cas contraire cela provoque une erreur de syntaxe. L'utilisation de la fonction s peut aider dans les cas où une chaîne complexe est requise comme paramètre.
Exemples :	<ol style="list-style-type: none"> 1 if instr(v5,'ab')>0 then v5/, fi, 2 if instr(s('v1'),v5)>0 then v1, fi, 3 left(v18,instr(v18,'.))-1,

instruction conditionnelle

Voir : **if ... then ... else ... fi** commande

iocc	index d'occurrence
-------------	---------------------------

Support :	CISIS
Type de fonction :	Numérique
Syntaxe :	iocc
Définition :	Retourne le numéro d'ordre (index) de l'occurrence (en commençant à 1), ou 0 dans le cas contraire.
Exemples :	<ol style="list-style-type: none"> 1 ("Auteur : "v1/, ,if iocc > 3 then 'et al.',break, fi), 2 (f(iocc,3,0), . v10/),
Voir aussi :	nocc fonction

L

I(clé) I([fichier inversé],clé)	recherche de clé
--	-------------------------

Support :	Standard/CISIS
Type de fonction :	Numérique
Syntaxe :	I (<format clé>) I ([<format fichier inversé>]<format clé>) I ->ifname(<format clé>) (compatible avec le logiciel Winisis de l'Unesco)

l(clé) l([fichier inversé],clé)	recherche de clé
Définition :	Recherche dans le fichier inversé courant le terme généré par <format clé> (s'il existe) et retourne le NFM indiqué par le premier pointeur. Il est également possible de rechercher dans un autre fichier inversé dont le nom est indiqué par <format fichier inversé>.
Notes :	Les clés sont converties en majuscules avant la recherche de l'expression. Le mode d'affichage par défaut est mpl. Si un autre mode est spécifié dans le fichier FST, il faut en tenir compte dans <format clé> pour que la clé soit générée. Si la clé n'est pas trouvée, la fonction retourne zéro. Le paramètre <format fichier inversé> doit générer une chaîne avec un nom valide de fichier inversé, autrement cela provoque une erreur de syntaxe. Cette fonction est également souvent utilisée en même temps que la fonction ref pour permettre la visualisation des champs d'un autre enregistrement.
Exemples :	<ol style="list-style-type: none"> 1 if l(v15)<> 0 then Terme : v15, fi, 2 ref(l(['bibl']v1,'-',v2),v10/), 3 ref->bibl(l->bibl(v1,'-',v2),v10/),
Voir aussi :	ref fonction

left(chaine,longueur)	extraction d'une chaîne de caractères à gauche
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	left (<format-1>,<format-2>)
Définition :	Retourne une nouvelle chaîne qui contient les <i>n</i> premiers caractères de la chaîne de départ <format-1>, en commençant à partir de la gauche, le nombre de caractères étant déterminé par <format-2>.
Notes :	Si la valeur de la chaîne générée par <format-2> est plus grande que la taille définie par <format-1>, la fonction retourne la chaîne <format-1>. Si <format-2> est zéro ou un nombre négatif, la fonction retourne une chaîne NULLE.
Exemples :	<ol style="list-style-type: none"> 1 if left(v1^n,2)='Ma' then v1^n/, fi, 2 left(v1,instr(v1,'.')->1),
Voir aussi :	right fonction mid fonction

littéral conditionnel

Voir : **"chaîne"** commande

littéral conditionnel répétitif

Voir : **|chaîne|** command

littéral inconditionnel

Voir : **'chaîne'** commande

lw(nombre)**définition de la largeur de ligne**

Support : CISIS
 Type de fonction : Numérique
 Syntaxe : **lw(<entier>)**
 Définition : Définit la largeur de la ligne de sortie à <entier> caractères.
 Notes : La largeur par défaut de la ligne de sortie est de 76 caractères.
 Exemples :
 1 **if size(v10) > 76 then lw(254), fi,**
 2 **lw(70),v20/,lw(10),v30/,**

M**mdl, mdu,
mhl, mhu,
mpl, mpu****mode d'affichage**

Support : Standard
 Syntaxe : **m<mode><conversion>**
 Définition : Définit un nouveau mode de visualisation pour la sortie actuelle.
 Notes : Le mode par défaut est mpl. <mode> représente le mode que l'on souhaite définir. <conversion> spécifie si l'on définit la conversion en majuscules. MODE peut apparaître plusieurs fois dans un format et son effet sur le format sera actif jusqu'à ce que l'on définisse un nouveau mode.
 <mode> peut avoir les paramètres suivants :
p = épreuve : les champs sont affichés comme ils ont été saisis dans les enregistrements.
h = en-tête : les caractères de contrôle et les délimiteurs de champs sont ignorés, en revanche les délimiteurs de sous-champs sont remplacés par des signes de ponctuation.
d = données : identique au mode en-tête, ajoute seulement un point à la fin du champ suivi de deux espaces.
 <conversion> peut avoir les options suivantes :
u : convertit les données en majuscules (*u* de *upper case*)
l : laisse les données en minuscules (*l* de *lower case*), en réalité laisse les données comme elles étaient saisies.
 Exemples :
 1 **mpl,"Premier auteur : "v10[1]/,**
 2 **mpu,"Deuxième auteur : "v10[2]/,**
 3 **mdl,"Troisième auteur : "v10[3]/,**

mfn mfn(longueur)	numéro de l'enregistrement
Support :	Standard
Type de fonction :	Chaîne de caractères ou numérique
Syntaxe :	mfn mfn(<entier>)
Définition :	Retourne le MFN (Master File Number) d'un enregistrement.
Notes :	Un nombre entier peut être passé comme paramètre pour définir la longueur de la chaîne que retournera la fonction MFN. mfn retourne une valeur de type numérique ou chaîne de caractères selon le format.
Exemples :	<ol style="list-style-type: none"> 1 'Enregistrement : ',mfn(3)/, 2 if mfn > 2 then mfn/, fi, 3 ref(mfn-1,v2/),
Voir aussi :	ref fonction l fonction
mid(chaîne,départ, longueur)	extraction d'une chaîne de caractères
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	mid(<format-1>,<format-2>,<format-3>)
Définition :	Retourne une nouvelle chaîne de caractères, contenant un nombre précis de caractères de la chaîne de départ (<format-1>). <format-3> spécifie le nombre de caractères à lire dans <format-1> et <format-2> la position à partir de laquelle on extrait les caractères de <format-1>.
Notes :	Si <format-2> est plus grand que la taille de <format-1>, la fonction retourne une chaîne NULLE. Si <format-2> est zéro ou un nombre négatif, la valeur par défaut est 1.
Exemples :	<ol style="list-style-type: none"> 1 mid(v2,2,80), 2 mid(v1,instr(v1,'clé'),size(v1))/,
Voir aussi :	right fonction left fonction
mode d'affichage	
<i>Voir :</i>	mdl, mdu, mhl, mh, mpl, mpu commande
mstname	nom du fichier maître
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	mstname
Définition :	Retourne le nom fichier maître actuel/courant.
Exemples :	1 'Base de données courante : ', mstname /,

mstname	nom du fichier maître
	2 ref(['noms'] (['noms']X39BJ), , 'La base de données est maintenant ', mstname /),

N

n	sélecteur de champ fictif (absent)
Support :	Standard
Syntaxe :	n <sélecteur de champ>
Définition :	Vérifie l'absence d'un champ. On l'utilise en l'associant à un littéral conditionnel.
Notes :	Comme sélecteur de champ fictif, il ne retourne aucune valeur.
Exemples :	1 "ce texte s'affiche si le champ 10 est absent"n10, 2 "Auteur : "v10/, , "Auteur : "n10,v20/,
Voir aussi :	" string " commande d commande v commande

newline(chaine)	insère / rétablit un caractère de nouvelle chaîne
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	newline (<format>)
Définition :	Insère et/ou rétablit la paire CR/LF par défaut avec le(s) caractère(s) générés par <format>.
Notes :	<format> peut aussi contenir une séquence d'échappement réservée comme : \r – est un saut de paragraphe \n – est un saut de ligne Les sauts de paragraphe et de ligne \ suivants seront automatiquement remplacés par la chaîne générée par <format> jusqu'à ce qu'un nouvel de la fonction newline définisse une nouvelle paire de chaînes (ou de caractères) pour un saut de ligne ou de paragraphe.
Exemples :	1 newline (if v151='unix' then '\n' else '\r\n' fi, 2 newline (v301), 3 newline (' '),
Voir aussi :	/ commande # commande

nocc(champ)	nombre d'occurrences
Support :	CISIS

nocc(champ) nombre d'occurrences

Type de fonction :	Numérique
Syntaxe :	nocc (<sélecteur de champ>)
Définition :	Retourne le nombre d'occurrences pour le champ ou le sous-champ spécifié par <sélecteur de champ>.
Notes :	Cette fonction n'accepte seulement comme paramètre que <sélecteur de champ> avec un champ ou un sous-champ. Tous les autres paramètres de la fonction <sélecteur de champ> provoquent une erreur de syntaxe s'ils sont utilisés.
Exemples :	<ol style="list-style-type: none"> 1 if nocc(v3)> 10 then 'Les occurrences sont en trop grand nombre.'/, fi, 2 'Il y a ',f(nocc(v20),2,0),' auteurs.'/,
Voir aussi :	iocc fonction v commande

nom du fichier maître

Voir : **mstname** fonction

npost(clé) pointeurs correspondant à une clé
npost([fichier inversé],clé)

Support :	CISIS
Type de fonction :	Numérique
Syntaxe :	npost (<format clé>) npost ([<format>],<format clé>)
Définition :	Retourne tous les pointeurs d'une clé d'un fichier inversé (la clé est générée par <format clé>). Si <format> est précisé, cela doit générer une chaîne contenant le nom du fichier inversé à utiliser. <format clé> génère la clé à rechercher dans le fichier inversé.
Exemples :	<ol style="list-style-type: none"> 1 if npost(v1)> 1 then 'clé en double ',v1,' found'/, fi, 2 'Il y a ',f(npost(v20),3,0),'clés pour ',v20,'.'/,
Voir aussi :	I fonction

nombre d'occurrences

Voir : **nocc** fonction

numéro de l'enregistrement

Voir : **mfn** fonction

P

p(sélecteur de champ)	vérification de la présence du champ
Support :	Standard
Type de fonction :	Booléen
Syntaxe :	p (<sélecteur de champ>)
Définition :	Retourne VRAI si le champ est présent, et FAUX dans le cas contraire.
Notes :	Peut utiliser tous les paramètres de la fonction v (sélecteur de champ) sauf l'indentation.
Exemples :	<ol style="list-style-type: none"> 1 if p(v12) then v12 else v13, fi, 2 if p(v50^a) and p(v50^b) then v50^a/,v50^b/, fi,
Voir aussi :	a fonction v commande

pointeurs correspondant à une clé

Voir : **npost** fonction

proc(format actualisation)	actualisation du champ
Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	proc (<format actualisation>)
Définition :	Ajoute ou remplace les champs dans l'enregistrement courant. <format actualisation> est un format qui génère les commandes spécifiques de mise à jour que la fonction doit exécuter.
Notes :	<p>Une spécification d'actualisation des champs est une chaîne de caractères composée des commandes d (supprimer), a (ajouter) et h (ajouter) et les modifications que celles-ci doivent exécuter. Les modifications sont appliquées à l'enregistrement courant. Toutes les commandes d (supprimer) doivent précéder les commandes a et h (ajouter).</p> <p>Spécification des commandes :</p> <p>d* - efface tous les champs de l'enregistrement</p> <p>d<étiquette de champ> - efface toutes les occurrences du champ <étiquette de champ></p> <p>d<étiquette de champ>/<occ> - efface l'occurrence <occ> du champ <étiquette de champ></p> <p>A<étiquette de champ>#<chaîne># - ajoute <chaîne> comme une nouvelle occurrence du champ <étiquette de champ></p> <p>h<étiquette de champ> <n> <chaîne> - ajoute <chaîne>, de <n> bytes de long, comme une nouvelle occurrence du champ <étiquette de champ></p> <p>Le délimiteur # peut être un caractère non numérique. Il doit y avoir un espace entre les paramètres <étiquette de champ>, <n> et <chaîne> de la commande h.</p>
Exemples :	<ol style="list-style-type: none"> 1 proc('d70', a10# v70 #),

proc(format actualisation)	actualisation du champ
-----------------------------------	-------------------------------

2 **proc**(if v24*0.4 = 'Tech' then 'd*', fi),

putenv(expression)	définition d'une variable d'environnement
---------------------------	--

Support : CISIS
 Type de fonction : Chaîne de caractères
 Syntaxe : **putenv**(<format>)
 Définition : Définit une variable d'environnement au système d'exploitation avec sa valeur correspondante.
 Notes : La variable ne sera valable que pendant la durée du processus en cours.
 Exemples :
 1 **putenv**('TEST=test'),getenv('TEST'),
 2 set CIPAR=unfichier
 set
 mx null "pft=**putenv**('CIPAR=unautre'),getenv('CIPAR')/"
 set
 Voir aussi : **getenv** fonction

R

ravr(chaîne)	valeur moyenne d'une expression
---------------------	--

Support : Standard
 Type de fonction : Numérique
 Syntaxe : **ravr**(<format>)
 Définition : Retourne la valeur moyenne d'un format donné. <format> doit générer une expression alphanumérique.
 Notes : Peut être utilisé pour calculer la moyenne des valeurs numériques contenues dans des champs répétitifs.
 Exemples :
 1 f(**ravr**(s(v8,x1,v1)),3,0),
 2 f(**ravr**(v1,x1,v2),5,2),
 3 f(**ravr**('8;15;16.73'),3,2),
 4 if **ravr**(v20|;|)>=5 then 'pass'/ else 'fail'/, fi,
 Voir aussi : **rmin** fonction
rmax fonction
rsum fonction

ref(mfn, format) ref([fichier maître]mfn,format)	exécution d'un format dans un enregistrement associé
---	---

Support : Standard/CISIS
 Type de fonction : Chaîne de caractères

ref(mfn, format) ref([fichier maître]mfn,format)	exécution d'un format dans un enregistrement associé
---	---

Syntaxe :	ref (<expression>,<format>) ref ([<format dbnom>]<expression>,<format>) ref->dbnom (<expression>,<format>) (compatible avec le logiciel Winisis de l'Unesco)
Définition :	Exécute <format> dans l'enregistrement associé au moyen de <expression>. Si on précise <format dbnom> on peut faire référence à une autre (ou à la même) base de données.
Notes :	<expression> peut correspondre à n'importe quel format qui retourne le MFN d'un enregistrement. La fonction I peut être utilisée pour exécuter une recherche et retourner le MFN du premier enregistrement trouvé.
Exemples :	<ol style="list-style-type: none"> ref(I(v3),v1/,v2/,v3/), if ref(['compte']I(['utilisateur']v2),v4)='actif' then Nom : v10/, fi, (if p(v99) then ref([v99]1,v30/), fi), ref->bibl(I->bibl(v1,'-',v2),v10/),
Voir aussi :	I fonction

recherche de chaîne de caractères
--

<i>Voir :</i>	instr fonction
---------------	-----------------------

recherche de clé

<i>Voir :</i>	I fonction
---------------	-------------------

replace(chaîne1, chaîne2, chaîne3)	remplacement d'une chaîne de caractères
---	--

Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	replace (<format-1>,<format-2>, <format3>)
Définition :	Retourne une nouvelle chaîne de caractères en remplaçant <format-2> par <format-3> dans <format-1>.
Notes :	<p>Si <format-2> est une chaîne de caractères nulle ou si elle ne se trouve pas dans <format-1>, la fonction retourne la chaîne <format-1>.</p> <p>Si <format-3> est une chaîne de caractères nulle, la chaîne <format-2> sera supprimée de <format-1>.</p> <p>Replace est une option sensible pour deux chaînes de caractères : la chaîne à rechercher (<format-2>) et la chaîne de remplacement (<format-3>).</p>
Exemples :	<ol style="list-style-type: none"> replace('Marie Et Jean','Et','et')/, if replace(v1^a,'01x','01X')='894501X' then v1^n/, fi,

replace(chaine1, chaine2, chaine3)	remplacement d'une chaîne de caractères
---	--

- | | |
|---|--|
| 3 | replace(s(v304,v333),',', ' '),/ |
| 4 | replace (s(if v415='espagnol' then v299 else 'aucun' fi),v1,v759)/, |

retour à la ligne conditionnel

<i>Voir :</i>	/ commande
---------------	------------

retour à la ligne inconditionnel

<i>Voir :</i>	# commande
---------------	------------

right(chaine, longueur)	extraction d'une chaîne de caractères à droite
--------------------------------	---

Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	right (<format-1>,<format-2>)
Définition :	Retourne une nouvelle chaîne qui contient les <i>n</i> derniers caractères de la chaîne de départ <format-1>, en commençant à partir de la droite, le nombre de caractères étant déterminé par <format-2>.
Notes :	Si la valeur de la chaîne générée par <format-2> est plus grande que la taille définie par <format-1>, la fonction retourne la chaîne <format-1>. Si <format-2> est zéro ou un nombre négatif, la fonction ne retourne rien.
Exemples :	<ol style="list-style-type: none"> 1 if right(v1^n,1) = 'r' then v1^n/, fi, 2 right(v65,4)/,

rmax(chaine)	valeur maximum d'une expression
---------------------	--

Support :	Standard
Type de fonction :	Numérique
Syntaxe :	rmax (<format>)
Définition :	Retourne la valeur maximale d'un format donné. <format> doit générer une chaîne de caractères.
Notes :	Peut être utilisé pour calculer le maximum des valeurs numériques contenues dans un champ répétitif.
Exemples :	<ol style="list-style-type: none"> 1 f(rmax('72,54,2'),2,0), 2 f(rmax(v1,x1,v4,x1,(v8 ,)),5,2), 3 if rmax(v40 ;)>val(v41) then 'Limite de ',v41,'dépassée.'/, fi,
Voir aussi :	rmin fonction ravr fonction rsum fonction

rmin(chaine)	valeur minimum d'une expression
---------------------	--

Support :	Standard
Type de fonction :	Numérique
Syntaxe :	rmin (<format>)
Définition :	Retourne la valeur maximale d'un format donné. <format> doit générer une chaîne de caractères.
Notes :	Dans le même genre que la fonction rmax , rmin peut calculer le minimum des valeurs numériques contenues dans un champ répétitif.
Exemples :	<ol style="list-style-type: none"> 1 f(rmin('10;2;5;4;-2'),2,0), 2 f(rmin(v1,x1,v2,x1,'44'),4,2), 3 if rmin(v80 ,v90 ,v100) < 1990 then 'Décennie inexacte.'/, fi,
Voir aussi :	rmax fonction ravr fonction rsum fonction

rsum(chaine)	somme d'un format
---------------------	--------------------------

Support :	Standard
Type de fonction :	Numérique
Syntaxe :	rsum (<format>)
Définition :	Retourne la somme d'un format donné. <format> doit générer une chaîne de caractères.
Notes :	Dans le même genre que les fonctions rmax et rmin , rsum calcule la somme des valeurs numériques contenues dans un champ répétitif.
Exemples :	<ol style="list-style-type: none"> 1 f(rsum('102,45,-37'),2,0), 2 f(rsum(v1,x1,v3,x1,f(val(v8)+2)),4,2), 3 if rsum(v20^d)>1000 then 'Interrompu.'/ else 'OK'/, fi,
Voir aussi :	rmax fonction ravr fonction rmin fonction

S

s0 ... s9	définition de variables - chaînes de caractères
------------------	--

Support :	CISIS
Type de fonction :	Chaîne de caractères
Syntaxe :	s <n>:=(<format>)
Définition :	CISIS définit 10 chaînes de variables, de s0 .. s9.

	Les variables sont initialisées comme des chaînes nulles à chaque fois que le format est exécuté.
Notes :	Les parenthèses autour du format sont obligatoires. Une variable – chaîne peut être utilisé, soit comme un opérande, soit comme une commande de format.
Exemples :	s1:=(‘CDS/ISIS’) s3:=(v10)
Voir aussi :	e0 .. e9 commande f fonction

s(expression) concaténation de chaînes de caractères

Support :	Standard
Type de fonction :	Chaîne de caractères
Syntaxe :	s (<format>)[paramètres]
Définition :	Retourne la concaténation de chaînes de caractères générées par <format>.
Paramètres :	extraction
extraction:	Extrait une partie du contenu d'une chaîne de résultat. <déplacement> indique la première position du caractère à extraire, tandis que <longueur> indique le nombre de caractères à extraire. Si <longueur> n'est pas précisé ou s'il est plus grand que la longueur réelle du champ, cela correspond par défaut à la fin de la chaîne.
Notes :	Peut être utilisé par des fonctions qui requièrent une chaîne de caractères comme paramètre.
Exemples :	1 if s (v1,v2,v3):'ABCDE' then s (v1,v2,v3)*0.50, fi, 2 if s (* v5 *): s (*E*)then 'Anglais'/, fi,
Voir aussi :	v commande

seconds calcul du nombre de secondes

Support :	CISIS
Type de fonction :	Numérique
Syntaxe :	seconds(<format>)
Définition :	Fonction numérique pour calculer le nombre de secondes depuis 1 ^{er} janvier 1970 00:00:00 jusqu'à présent ; elle est générée par < format> avec certains des formats suivants 'aaaammdd' 'aaaammdd hh' 'aaaammdd hhmm' 'aaaammdd hhmmss'
Notes :	La date générée par <format> doit se trouver dans l'intervalle '19700101 000000' à '20380118 031407' Un jour contient 24 x 60 x 60 = 86400 secondes
Exemples :	mx null "pft=date/f(seconds(date) – seconds(20010305 172915'),1,0)" <i>va donner</i>

seconds	calcul du nombre de secondes
----------------	-------------------------------------

20010305 172916 1 63
 1..
 20010305 172916 1 63
 2..
 20010305 172916 1 63
 3..

Voir aussi : **date** commande
datex fonction

select ... case ... elsecase ... endsel	exécution conditionnelle en bloc
--	---

Support : CISIS

Syntaxe : **select** <format expression>
case <option-1>: <format-1>
case <option-2>: <format-2>
case <option-n>: <format-n>
[**elsecase** <format-0>]
endsel

Définition : Évalue <format expression> et compare le résultat avec chaque option **case** (<option-1>, <option-2>...<option-n>). Si une option coïncide avec con <format expression>, le bloc d'instructions approprié est exécuté ; si aucune option ne correspond à <format expression> la clause **elsecase** (<format-0>) est exécutée, si elle a été définie.

Notes : <format expression> doit générer une chaîne de caractères ou une valeur numérique. Si <format expression> génère une chaîne, toutes les valeurs des options des clauses **case** doivent être du type chaîne de caractères. Si <format expression> génère une valeur numérique, les valeurs de l'option doivent aussi être de type numérique.

Exemples :

```

1 select s(v5)
  case '1': ,f(val(v5)/2,2,2)/,
  case '2': ,v5/,
  case '3': ,v6,'-',v1/,
  elsecase ,|Erreur dans le champ v5 = |v5/,
  endsel,

2 select nocc(v7)
  case 0: 'absent'/,
  case 1: 'une occurrence'/,
  case 2: 'deux occurrences'/,
  elsecase 'plus de 2 occurrences'/,
  endsel,
  
```

Voir aussi : **if ... then ... else ... fi** commande

sélecteur de champ

Voir : **v** commande

sélecteur de champ fictif (absent)*Voir :* **n** commande**sélecteur de champ fictif (présent)***Voir :* **d** commande**size(chaîne)****taille de la chaîne de caractères**

Support : CISIS

Type de fonction : Numérique

Syntaxe : **size**(<format>)

Définition : Retourne la taille d'une chaîne de caractères.

Notes : <format> doit retourner une chaîne de caractères, dans le cas contraire une erreur de syntaxe se produit.

Exemples :
1 **if size(v10) > 76 then lw(254), fi,**
2 **f(size(v10,v20),1,0),****somme d'un format***Voir :* **rsum** fonction**suppression de lignes vides***Voir :* % commande**system(expression)****exécution d'une commande de système d'exploitation**

Support : CISIS

Type de fonction : Chaîne de caractères

Syntaxe : **system**(<format>)

Définition : Exécute l'argument généré par <format> comme une commande de système d'exploitation.

Notes : <format> doit générer une chaîne de caractères qui contient la commande à exécuter. La sortie éventuelle générée par cette commande sera envoyée directement à la sortie standard du système (par défaut, l'écran).

Exemples :
1 **system('dir'),**
2 **if p(v2) then system('type ',v2), fi,****T****taille d'une chaîne de caractères***Voir :* **size** fonction

type de chaîne

Voir : **type** fonction

type(chaîne)**type de chaîne**

Support : CISIS

Type de fonction : Chaîne de caractères

Syntaxe : **type(<format>)**

Définition : Retourne le type d'une chaîne de caractères de la manière suivante :
 A - si la chaîne contient seulement des caractères alphabétiques (conformes à la table de caractères alphabétique par défaut, comme ISISAC.TAB) ou des espaces.
 N - si la chaîne contient seulement des chiffres (0-9)
 X – dans les autres cas.

Notes : <format> doit générer une chaîne de caractères, sinon cela provoque une erreur de syntaxe.

Exemples :

- 1 if **type(v1)**='N' then f(val(v1),3,2)/ else v1/, fi,
- 2 if s(**type(v1),type(v2),type(v3)**)<>'AAA' then 'Type de caractère invalide détecté'/, fi,

V**v****sélecteur de champ**

Support : Standard

Syntaxe : **v**<étiquette de champ>[paramètres]

Définition : Génère le contenu des champs. Le contenu peut être sélectionné, restreint, extrait ou faire l'objet d'une indentation au moyen de paramètres de la même commande (voir plus bas). **v** veut dire champ de longueur variable.

Paramètres : Sous-champ, occurrence, extraction et indentation.

ordre syntaxique : ^<sous-champ> [<index>[.<index supérieur>]]
 *<déplacement>.<longueur> (<première ligne>,<ligne suivante>)

sous-champ : Restreint la sortie au contenu d'un sous-champ. Si le champ existe mais que le sous-champ est absent, aucune sortie n'est générée.

v	sélecteur de champ
occurrence :	Restreint la sortie à une ou plusieurs occurrences d'un champ répétitif. <index> et <index supérieur> se réfèrent respectivement à la première (ou l'unique) et aux occurrences suivantes. Si le <index> spécifié est plus grand que le nombre actuel d'occurrences, aucune sortie n'est générée. De même lorsque le champ n'est pas répétitif ou lorsque <index> correspond à un nombre égal ou plus grand que 2. Cependant, si <index> a la valeur 1 et s'il est utilisé dans un champ non répétitif, le contenu est affiché normalement. Ce paramètre doit être utilisé en dehors d'un groupe répétitif, sinon <index supérieur> est ignoré. Si le double point (..) est employé et <index supérieur> manque LAST est appliqué par défaut. Le mot clé LAST prend la valeur du nombre total d'occurrences.
extraction:	Extrait une partie du contenu d'un champ, sous-champ ou occurrence. <déplacement> indique la première position du caractère à extraire, tandis que <longueur> indique le nombre de caractères à extraire. Si <longueur> n'est pas précisé ou s'il est plus grand que la longueur réelle du champ, cela correspond par défaut à la fin du champ.
indentation :	Aligne la sortie du champ, du sous-champ, de l'occurrence, du contenu extrait, selon <première ligne> (alignement de la première ligne) et <ligne suivante> (alignement des lignes suivantes). Les deux valeurs sont des constantes numériques. Si la position de la ligne courante diffère de zéro, l'indentation est désactivée.
Notes :	Le fonctionnement de la commande v dépend des paramètres utilisés. Aucune sortie n'est générée lorsque le champ est absent ou lorsque le paramètre exécute une restriction ou une extraction qui se trouve hors des limites.
Exemples :	<ol style="list-style-type: none"> 1 v2/,v3^a - ,v1/, 2 v1^n*0.3, 3 (; +v3^s)/, 4 v20[4], 5 v10[2..7]/, 6 v5[3..]/,/* equivaut à ,v5[3..LAST], */ 7 v1[LAST]*2.7/, 8 v1(5,5)/, 9 Titre : v1^n(5,5)/,
Voir aussi :	<p>"chaîne" [littéral conditionnel] d [sélecteur de champ fictif] n [absent] chaîne [littéral conditionnel répétitif] (format) [groupe répétitif]</p>

val(chaine)	conversion d'une chaîne de caractères en valeur numérique
Support :	Standard
Type de fonction :	Numérique
Syntaxe :	val (<format>)
Définition :	Retourne la valeur numérique de l'argument généré par <format>.
Notes :	Si <format> produit seulement des caractères non numériques, la fonction retourne zéro. Si <format> trouve plus d'une valeur numérique, seule la première sera retournée.
Exemples :	<ol style="list-style-type: none"> 1 if val(v2)>5 then 'Erreur'/ else 'OK'/, fi, 2 f(val(v2)/3,4,2),

valeur maximum d'une expression

Voir : **rmax** fonction

valeur minimum d'une expression

Voir : **rmin** fonction

valeur moyenne d'une expression

Voir : **ravr** fonction

W

while **contrôle conditionnel d'exécution**

Support :	CISIS
Syntaxe :	while <condition> (<format>) <condition> est une expression booléenne. <format> est un format CISIS tant que l'expression booléenne retourne VRAI.
Définition :	La commande WHILE permet d'exécuter de manière répétitive un format (boucle).
Notes :	Si la valeur initiale de la condition retourne FAUX, le format n'est pas exécuté. Pour que l'exécution se termine, le format doit inclure une commande qui permettra de convertir la condition en FAUX. Dans le cas contraire, cela génère une boucle sans fin et le système ne répondra plus à l'utilisateur.
Exemples :	<pre>e1:=1, e2:=nocc(v16), while e1<=e2 (f(e1,1,0),'. ', v16[e1]/</pre>

while **contrôle conditionnel d'exécution**

```
e1:=e1+1,
```

```
),
```

L'exemple affiche chaque occurrence du champ v16 (auteur) précédé du numéro de l'occurrence.

1. Premier auteur
2. Deuxième auteur
3. Troisième auteur

Voir aussi :

if ... then ... else ... fi commande

select ... case ... elsecase ... endsel commande

X**x** **insertion d'espaces**

Support : Standard

Syntaxe : **x**<entier>

Définition : Insert un nombre d'espaces défini par <entier>.

Notes : Si <entier> est plus grand que l'espace disponible dans la ligne en cours, passe à la ligne suivante.

Exemples : 1 'Nom : ',**x5**,v1^n/,

2 (v1,**x3**,v2,**x8**,v3/),

Voir aussi : **c** commande

Références bibliographiques

1. UNESCO. *Mini-micro CDS/ISIS: Reference manual* (version 2.3). Organized by Giampaolo Del Bigio. Paris: United Nations Educational, Scientific and Cultural Organization, 1989. 286 p. ISBN 92-3-102-605-5.
2. BUXTON, Andrew, HOPKINSON, Alan. *The CDS/ISIS for Windows Handbook* [online]. Paris: United Nations Educational, Scientific and Cultural Organization, 2001 [cited 30 August 2006]. 164 p. Available from internet: <<http://bvsmodelo.bvs.br/download/winisis/winisis-handbook-en.pdf>>.
3. SUTER, Tito. "Prehistoria" e historia del MicroISIS [online]. In: *Manual para instructores de Winisis*. Buenos Aires: Centro Atómico Constituyentes (CAC), Comisión Nacional de Energía Atómica (CNEA), 1999 [citado el 30 Agosto 2006]. p. 21-26. Disponible en internet: <<http://www.cnea.gov.ar/cac/ci/isis/isidams.htm>>.

Glossaire

- **Base de données.** Collection de données structurée pour qu'il soit possible d'y accéder et de les manipuler facilement. Est formée d'unités d'information appelées *enregistrements* dont les différents éléments sont représentés par des champs et des sous-champs. Par exemple, dans un "fichier clients", chaque client représente un enregistrement possédant plusieurs champs, comme "NOM", "CODE DU CLIENT", "TÉLÉPHONE" etc.
- **Bases de données bibliographiques.** Version électronique d'un catalogue ou d'un index bibliographique.
- **CDS/ISIS - MicroISIS.** Logiciel développé et maintenu par l'UNESCO pour le traitement des données bibliographiques.
- **Champ.** Élément d'un enregistrement permettant de stocker une information spécifique. Voir *Base de données*.
- **Clé.** Expression identifiant une ou plusieurs informations d'un type déterminé et pouvant être utilisée en recherche.

- **Enregistrement.** Ensemble structuré de données qui permet de stocker tous les éléments de données d'une unité d'information. Voir *Base de données*.
- **Fichier.** En informatique, un ensemble de données qui peut être enregistré dans un certain dispositif de stockage. Les fichiers de données sont créés par des applications, comme par exemple un traitement de textes.
- **Fichier inversé.** Ensemble de six fichiers physiques, dont cinq contiennent le dictionnaire des termes de recherche (organisés en arbre B*) et le sixième contient la liste des pointeurs associés à chaque terme. Afin d'optimiser le stockage sur disque, les deux arbres B* sont séparés et gèrent, l'un les termes d'au plus 10 caractères (stockés dans les fichiers *.N01* et *.L01*) et l'autre les termes de plus de 10 caractères (stockés dans les fichiers *.N02* et *.L02*). Le fichier *.CNT* contient les champs de contrôle des deux arbres B*. Dans chaque arbre B*, les fichiers *.NOx* contiennent les noeuds de l'arbre et les fichiers *.LOx* contiennent les feuilles. Les enregistrements des feuilles pointent à l'endroit où se trouvent les pointeurs qui contiennent les informations pour localiser les enregistrements (postings) dans la base de données. Ce fichier a une extension *.IFP*.
- **Format d'affichage.** Ensemble de commandes qui déterminent comment doit être la sortie des données d'une base de données ISIS.
- **Format électronique.** Toute forme de stockage, récupération et présentation d'information susceptible de faire l'objet d'une transmission en ligne ou d'un enregistrement sur des supports magnétiques ou optiques.

- **Format ISO (échange de données).** Format standard défini par l'ISO pour l'échange de données entre institutions, réseaux et utilisateurs. Se réfère à la norme ISO 2709.
- **Format LILACS.** Format de description bibliographique défini par BIREME, basé sur le *Reference Manual for Machine-readable Bibliographic Descriptions* de l'UNISIST.
- **Indexation.** Procédure pour identifier et décrire le contenu d'un document avec des termes représentant les sujets correspondant à ce document, dans le but de le retrouver ultérieurement.
- **Pointeur.** Adresse d'une clé extraite du fichier maître.
- **Sauvegarde.** Procédure par laquelle un ou plusieurs fichiers sont dupliqués sur un autre dispositif de stockage (bande ou disque), pour produire une copie de sécurité qui pourrait être restaurée si certaines données sont accidentellement effacées ou s'il se produit des dommages physiques affectant les données originales.
- **Sous-champ.** Élément qui contient la plus petite partie de l'information dans un champ, et qui peut n'avoir aucune signification si elle est considérée de manière isolée.
- **UNISIST.** Programme intergouvernemental relatif à la coopération dans le domaine de l'information scientifique et technologique.