

Guide de l'Utilisateur de Generator Manager

Version 8.1.3

Novembre 2013



IKAN Solutions N.V.
Kardinaal Mercierplein 2
B-2800 Malines
BELGIQUE

Copyright © 2013, IKAN Solutions N.V.

Aucune partie de ce document ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans l'autorisation explicite et écrite de IKAN Solutions N.V.

MetaSuite est une marque déposée de IKAN Solutions N.V.

Table des Matières

Chapter 1 - À propos de ce manuel.....	1
1.1. Publications connexes	1
1.2. À propos de MetaSuite	2
Chapter 2 - Se connecter	3
Chapter 3 - Créer un Dictionnaire/ Saisir la Clé de licence	6
3.1. Créer un Fichier de Dictionnaire	6
3.2. Saisir une Clé de licence	9
Chapter 4 - L'écran Options du Dictionnaire	12
Chapter 5 - Écran de génération - Aperçu	15
Chapter 6 - Générer - Implémenter les instructions MIL.....	18
6.1. L'écran "Générer pour MIL"	19
6.2. Copier des Fichiers MIL existants.....	20
6.3. Ajouter des instructions MIL en utilisant l'Assistant Commandes.....	21
6.4. Référence des options CHANGE	24
CONSTRUCTION.....	25
FORMAT DE DATE	25
DATE BASCULE SIÈCLE	26
DÉCIMALE	26
EXEC	27
INTER-PONCTUATION	27
NULL	28
NULLABLE	28
PAGE	29
GUILLEMETS.....	29
SQL	30
SQL-QUOTE	31
DYNAMIQUE	31
LANGAGE D'EXPORT	32
UNICODE-CCSID.....	32
CARACTÈRE-CCSID	32

6.5. Options de TABLE - Référence	33
<i>LIST</i>	33
<i>ADD</i>	34
<i>COPY</i>	35
<i>DELETE</i>	35
<i>EXPORT</i>	35
<i>IMPORT</i>	36

Chapter 7 - Générer les fichiers MDL 37

7.1. L'écran "Générer - MDL"	38
7.2. Éditer les Fichiers MDL existants	39
7.3. Utiliser les Outils MDL - Générer Cible MDL en délimité	40
7.4. Utiliser les Outils MDL - Générer Cible MDL en XML	42
7.5. Utiliser les Outils MDL - Analyser la Source XML	45

Chapter 8 - Générer les fichiers MSM..... 47

Chapter 9 - Générer les fichiers MXL..... 49

9.1. Générer un Fichier MXL.....	50
9.2. Générer plusieurs Fichiers MXL en mode batch.	52

Chapter 10 - Maintenance des tables techniques - Aperçu 54

Chapter 11 - Maintenance des tables techniques - Tables MGL 58

11.1. L'écran de Maintenance des tables techniques MGL	59
11.2. Tables MGL: Description détaillée	64
11.3. Tables de Code-Contrôle	65
<i>Format</i>	67
<i>Tables de Code-Contrôle Source</i>	67
<i>Tables de Code-Contrôle Cible</i>	71
<i>Tables de Code-Contrôle de l'Utilisateur</i>	73
11.4. Tables de code COBOL.....	74
11.5. Exemples de table de code.....	74
<i>Exemple 1 - Table de Code pour appeler des Sous-programmes E/S</i>	74
<i>Exemple 2 - Table de Code pour appeler des Sous-programmes de reformatage</i>	75
11.6. Créer des Tables de Code-Contrôle personnalisées	76
11.7. Créer des Tables de code COBOL personnalisées	76

Chapter 12 - Maintenance des tables techniques - Tables MRL..... 78

12.1. L'écran de Maintenance des tables techniques MTL	79
--	----

Chapter 13 - Les autres fonctions..... 84

13.1. Paramètres initiaux du Générateur.....	84
--	----

13.2. Parcourir les dossiers	85
------------------------------------	----

Chapter 14 - Commandes du langage d'installation..... 88

14.1. NEW.....	89
<i>Colonnes 1-3.....</i>	89
<i>nnnn.....</i>	89
14.2. CHANGE DEFAULT BUILD.....	89
<i>Numéro de construction.....</i>	89
<i>Exemple.....</i>	89
14.3. CHANGE DEFAULT DATE	89
<i>ISO.....</i>	90
<i>EUR.....</i>	90
<i>JIS.....</i>	90
<i>USA.....</i>	90
<i>(siècle, année-bascule).....</i>	90
14.4. CHANGE DEFAULT DECIMAL	90
14.5. CHANGE DEFAULT EXEC.....	91
14.6. CHANGE DEFAULT INTERPUNCTION.....	91
<i>OFF.....</i>	91
<i>NO.....</i>	91
<i>YES.....</i>	92
14.7. CHANGE DEFAULT NULL.....	92
<i>Caractère Null.....</i>	92
14.8. CHANGE DEFAULT NULLABLE	92
<i>L'option Nullable</i>	92
14.9. CHANGE DEFAULT PAGE	93
14.10.CHANGE DEFAULT QUOTE	93
14.11.CHANGE DEFAULT SQL DIALECT	94
14.12.CHANGE DEFAULT SQL-QUOTE	94
14.13.CHANGE DEFAULT DYNAMIC	94
14.14.CHANGE DEFAULT CHARACTER-CCSID.....	94
14.15.CHANGE DEFAULT UNICODE-CCSID	94
14.16.ADD TABLE	95
<i>Nom de table.....</i>	95
<i>La taille d'entrée.....</i>	95
<i>Entrée de table.....</i>	95
14.17.COPY TABLE	95
<i>Nom de table.....</i>	96
14.18.DELETE TABLE	96
<i>Nom de table.....</i>	96
14.19.LIST DEFAULT	96
<i>Exemple.....</i>	96
14.20.LIST TABLE	96
<i>Nom de table.....</i>	97

Exemple	97
14.21. LIST VERSION	97
Exemple	97
14.22. REMARKS	98

Chapter 15 - Langage de modèle MetaSuite 99

15.1. Variables, Expressions et Fonctions MTL	99
Variables MTL	99
Les variables MTL temporaires	100
Expressions MTL	101
Fonctions MTL	102
15.2. Variables du Générateur	104
15.3. Commandes MTL	105
Conventions de notation	106
Boucles FOR-NEXT	107
CALL table	108
DUMP	109
EXIT Table	109
FREEZE et UNFREEZE	109
GOTO	110
IF imbriqué	111
IMPLEMENT	112
REMARK	112
SET	112
SKIP	113
Structures EVALUATE	114
Structures IF	115
TRACE	116
Traitement des messages d'erreur	117
UNSET	120
15.4. Variables système MTL	121
15.5. Variables MTL spécifiques au Fichier Source	123
15.6. Variables MTL liées à l'Enregistrement Source	124
15.7. Variables MTL liées au Fichier Cible	125
15.8. Variables liées à l'Enregistrement Cible	126
15.9. Variables MTL liées au Champ	127

Chapter 16 - Langage d'exécution MetaSuite 131

16.1. Tables	131
16.2. Exemples	134
Windows	134
UNIX / Linux	135
VMS	136
OSD/BC BS2000	136
OS/390 Z/OS	137

Chapter 17 - Modules de chaînes de caractères 139

17.1. Description	139
17.2. Usage	140
17.3. INITSTR - Initialise les Champs de paramètres.....	140
Description	140
Entrée	140
Résultats	141
17.4. LENSTR - Définit la longueur d'une chaîne de caractères	141
Description	141
Entrée	142
Résultats	142
Remarques	142
Exemples	142
17.5. SUBSTR - renvoyer une chaîne de caractères à partir d'une chaîne de caractères	142
Description	143
Entrée	143
Résultats	143
Remarques	143
Exemples	144
17.6. SRCHSTR - Rechercher dans une chaîne de caractères	144
Description	144
Entrée	144
Résultats	145
Remarques	145
Exemples	145
17.7. REPLSTR - remplacer une sous-chaîne de caractères dans une chaîne de caractères.....	146
Description	146
Entrée	147
Résultats	147
Remarques	147
Exemples	148

Chapter 18 - Appeler le Générateur en mode batch 149

18.1. Préparation	149
18.2. Utiliser MSBGEN.EXE	149
18.3. Exemple	150

Chapter 19 - Messages du Générateur..... 152

19.1. Codes de retour du Générateur.....	152
19.2. Messages d'erreurs de syntaxe du Générateur	155
19.3. Messages d'erreur du Dictionnaire	156
19.4. La liste des messages du Générateur.....	157
Messages 001 à 050	157
Messages 051 à 100	160

Messages 101 à 150	163
Messages 151 à 200	167
Messages 201 à 250	169
Messages 251 à 300	172
Messages 301 à 350	174
Messages 351 à 400	177
Messages 401 à 450	179
Messages 451 à 500	182
Messages 501 à 550	184
Messages 551 à 600	186
Messages 601 à 650	189
Messages 651 à 700	193
Messages 701 à 711	199

À propos de ce manuel

Le Générateur de MetaSuite est une collection de blocs de construction COBOL, requis pour la traduction du Modèle MetaMap (MSM) en une application COBOL pour le Système d'exploitation demandé.

MetaSuite Generator Manager est un outil développé pour assister les administrateurs MetaSuite dans la génération et la manipulation du Dictionnaire du Générateur MetaSuite et des fichiers de chargement du Dictionnaire (fichiers DDL).

Le *Guide de l'Utilisateur MetaSuite Generator Manager* fait partie du jeu de documentation destiné aux administrateurs MetaSuite.

1.1. Publications connexes

La table suivante donne un aperçu complet du jeu de documentation MetaSuite.

Information sur la version	Description des fonctionnalités de la version 8.1.3
Guides d'installation	• Composant d'exécution BS2000/OSD • Composant d'exécution DOS/VSE • Composant d'exécution Fujitsu Windows • Composant d'exécution IBM COBOL • Composant d'exécution Micro Focus Windows • Composant d'exécution Micro Focus UNIX • Composant d'exécution OS/390 et z/OS • Composant d'exécution OS/400 • Composant d'exécution VisualAge Windows • Composant d'exécution VisualAge UNIX • Composant d'exécution VMS
Guides de l'Utilisateur	• Guide de l'Utilisateur d'INI Manager • Guide d'installation et de paramétrage • Guide d'introduction • Guide de l'Utilisateur de MetaStore Manager • Guide de l'Utilisateur de MetaMap Manager • Guide de l'Utilisateur de Generator Manager
Guides techniques	• Guide d'accès aux fichiers ADABAS • Guide d'accès aux fichiers IDMS • Guide d'accès aux fichiers IMS DLI • Guide d'accès aux fichiers RDBMS • Guide d'accès aux fichiers XML • Modules d'exécution • Guide de l'Utilisateur des fonctions définies par l'Utilisateur

1.2. À propos de MetaSuite

Si vous n'êtes pas familier de MetaSuite, la description technique suivante vous fournit un aperçu.

Le système MetaSuite	MetaSuite est conçu pour la récupération des données, l'extraction, la conversion et les rapports. Il comprend une interface utilisateur graphique basée sur un poste de travail et un composant d'exécution mainframe.
Interface de bases de données MetaSuite	MetaSuite peut accéder aux données à partir d'un certain nombre de systèmes de gestion de bases de données, en utilisant les mêmes commandes, la même structure de programme et les mêmes techniques d'extraction qui sont utilisées pour les fichiers non base de données. Chaque interface de base de données est disponible comme un enrichissement optionnel pour le produit de base.
MetaMap Manager	MetaMap Manager est l'outil utilisé pour la définition des Modèles. De tels Modèles sont construits intuitivement en décrivant les spécifications de programme globales, les définitions des fichiers d'entrée (données et processus) et celles des fichiers Cible (données et processus).
MetaStore Manager	MetaStore Manager est l'outil utilisé pour la maintenance des Métadonnées et les services de documentation.
Generator Manager	Generator Manager est l'outil d'administration de système. Toutes sortes de fonctionnalités de base et de possibilités de personnalisation sont prises en charge par cet outil.

Se connecter

1. Démarrez *MetaSuite Generator Manager*.

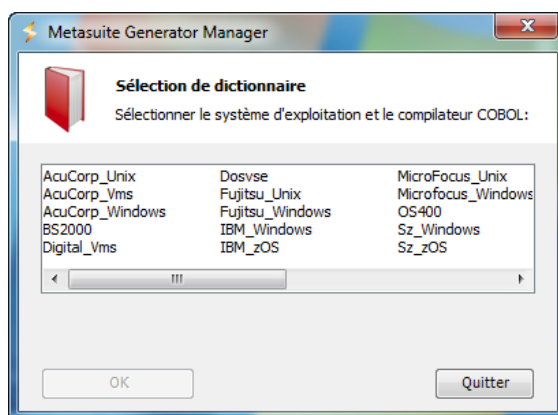
Sélectionnez "MetaSuite Generator Manager" à partir du menu rapide de MetaSuite ou cliquez deux fois sur le Fichier *GeneratorManager.exe* dans le dossier d'Installation MetaSuite.

2. Sélectionnez le Fichier INI requis s'il n'est pas trouvé automatiquement.

Si le Fichier d'initialisation par défaut *MetaSuite.ini* n'est pas trouvé dans le dossier d'Installation MetaSuite, le programme sera (ré)installé.

3. Si aucun Générateur par défaut n'est spécifié, vous devez sélectionner le Générateur MetaSuite requis.

Une liste déroulante avec tous les Générateurs MetaSuite installés dans cette configuration s'affiche.



Sélectionnez le Générateur requis, ou cliquez sur le bouton *Quitter* pour quitter le programme.

4. L'écran *MetaSuite Generator Manager* s'affiche.

Les icônes dans la colonne à gauche, vous permettent d'accéder aux différents écrans du Generator Manager.

Icône	Écran du Generator Manager
	Créer un Dictionnaire/ Saisir la Clé de licence (page 6)
	L'écran Options du Dictionnaire (page 12)
	Écran de génération - Aperçu (page 15)
	Maintenance des tables techniques - Aperçu (page 54)
	Les autres fonctions (page 84)

Icône	Écran du Generator Manager
	Sélectionner un autre Générateur
	Aide

Créer un Dictionnaire/ Saisir la Clé de licence

L'écran *Créer un Dictionnaire/ Saisissez la Clé de licence* est utilisé pour générer des Fichiers de Dictionnaire nouveaux ou existants, et pour saisir les Clés de licence.

Accédez à cet écran en cliquant sur le bouton *Créer un Dictionnaire/Saisir la Clé de licence* dans la colonne de gauche.

Pour une description détaillée, se référer aux procédures suivantes:

- [Créer un Fichier de Dictionnaire](#) (page 6)
- [Saisir une Clé de licence](#) (page 9)

3.1. Créer un Fichier de Dictionnaire

Utilisez cette fonction pour convertir des Fichiers de Dictionnaire de chargement (Fichiers DDL) en un Dictionnaire qui sera utilisé lors de la génération. Sans un tel Dictionnaire, aucune Source COBOL ne pourra être générée. En principe, cette opération fait partie du processus d'installation initial (Se référer au *Guide d'installation et de paramétrage*).

Plus tard, vous devrez également générer un Fichier de Dictionnaire pour enregistrer vos changements aux [paramètres MRL](#) (page 78) et aux [tables MGL](#) (page 58).

1. Accédez à l'écran *Créer un Dictionnaire/ Saisir la Clé de licence* en cliquant sur l'icône appropriée dans la colonne de gauche.

L'écran suivant s'affiche:

Créer un Dictionnaire

Saisissez les paramètres nécessaires pour générer un nouveau Dictionnaire et cliquez sur le bouton "Créer".

Taille du dictionnaire: 2500 Blocs de 4 Kilo-octets. Format de date: ISO

Longueur de page: 80 Dialecte SQL: Générateur par défaut

Largeur de page: 80 Mode d'exécution: (aucun)

Décimale: ☒ Point ☐ Virgule Année à deux chiffres comprise entre: 1951 - 2050

Guillemets: ☒ Single ☐ Double Langue: EN

Guillemets SQL: ☒ Single ☐ Double

Caractère Null: ?

Nullable: N

Inter-punctuation: OFF

☐ Importer les tables personnalisées

Saisissez la Clé de licence

Veuillez saisir le nom du client et la clé de licence, et appuyer sur le bouton "Appliquer".

Nom du client: IKAN Solutions

Clé de licence:

2. Saisissez les champs dans la section *Créer un Dictionnaire*.

Il contient les champs suivants:

Champ	Description
Taille du dictionnaire	Saisissez le nombre de blocs de la bibliothèque du Générateur à initialiser. Vous pouvez saisir une valeur plus grande (jusqu'à 9999) si vos transformations de données requièrent un traitement de Sources de données très grandes.
Longueur de page	Saisissez la longueur de page par défaut (en nombre de lignes) à utiliser pour les Rapports Cible si aucun paramètre de page n'est spécifié dans le Modèle MetaMap.
Largeur de page	Saisissez la largeur de page par défaut (en nombre de caractères) à utiliser pour les Rapports Cible si aucun paramètre de page n'est spécifié dans le Modèle MetaMap.
Décimale	Sélectionnez le caractère à utiliser comme Décimale. Vous pouvez choisir entre un <i>Point</i> et une <i>Virgule</i> .
Guillemets	Sélectionnez le caractère requis. Vous pouvez choisir entre <i>Simple</i> et <i>Double</i> .
Guillemets SQL	Sélectionnez le caractère de guillemet à utiliser pour les commandes SQL. Vous pouvez choisir entre <i>Simple</i> et <i>Double</i> .

Champ	Description
Caractère Null	Saisissez le caractère simple à utiliser pour indiquer la valeur Null dans un Champ d'un Fichier séquentiel. Le <i>Caractère Null</i> à utiliser pour le stockage INBOUND et OUTBOUND NULL dans un Champ. Dans le cas de INBOUND NULL, vous devez choisir le <i>Caractère Null</i> avec précaution vu que l'apparition du <i>Caractère Null</i> à la première position du champ détermine si le champ contient une valeur pour Null.
Nullable	Sélectionnez, à partir de la liste déroulante, l'option <i>Nullable</i> par défaut pour les Champs d'un Fichier séquentiel. Il y a deux options: - Sélectionnez <i>N</i> si vous voulez que tous les Champs sans paramètre spécifique pour DBNAME, soient traités comme étant NOT-NULLABLE (c.-à-d., DBNAME 'NOTNULL'). Le <i>Caractère Null</i> ne sera pas interprété pour ces Champs. - Sélectionnez <i>I</i> si vous voulez que tous les Champs sans paramètre spécifique pour DBNAME, soient traités comme étant INBOUND-NULLABLE (c.-à-d., DBNAME 'INNULL').
Inter-ponctuation	L'option "Inter-ponctuation" détermine le masque d'édition par défaut pour les Champs numériques. L'Utilisateur peut toujours écraser ce masque d'édition en sélectionnant un masque d'édition spécifique pour un Champ dans MetaStore Manager. Sélectionnez l'option d'inter-ponctuation requise à partir de la liste déroulante. Il y a trois possibilités: - Sélectionnez <i>OFF</i> si vous ne voulez ni de suppression de zéros ni d'inter-ponctuation. - Sélectionnez <i>NON</i> si vous voulez la suppression de zéros mais pas d'inter-ponctuation. - Sélectionnez <i>OUI</i> si vous voulez la suppression de zéros ainsi que l'inter-ponctuation.
Format de date	Sélectionnez, à partir de la liste déroulante, le format de date à utiliser pour les dates manipulées dans un environnement RDBMS. Les options suivantes sont disponibles: - Sélectionnez <i>ISO</i> ou <i>JIS</i> si les composants des dates doivent être séparés par un tiret (-). - Sélectionnez <i>EUR</i> si les composants des dates doivent être séparés par un point (.). - Sélectionnez <i>USA</i> si les composants d'une date doivent être séparés par une barre transversale (/).
Dialecte SQL	Sélectionnez, à partir de la liste déroulante, le dialecte SQL à utiliser lors de la génération de SQL imbriqué si aucun dialecte SQL spécifique n'est défini dans le Modèle MetaMap.
Mode d'exécution	Sélectionnez le mode d'exécution requis à partir de la liste déroulante. Les options suivantes sont disponibles: <i>IMS</i>: sélectionnez cette option pour permettre le redémarrage sous IMS/DC. <i>REDÉMARRABLE</i>: sélectionnez cette option pour permettre le redémarrage pour les autres environnements. - Sélectionnez (<i>aucun</i>) dans les autres situations.
Année à deux chiffres comprise entre	Ce champ permet de définir l'année de bascule, si le format de date n'inclut pas le siècle, comme par exemple dans le format de date YYMMDD.

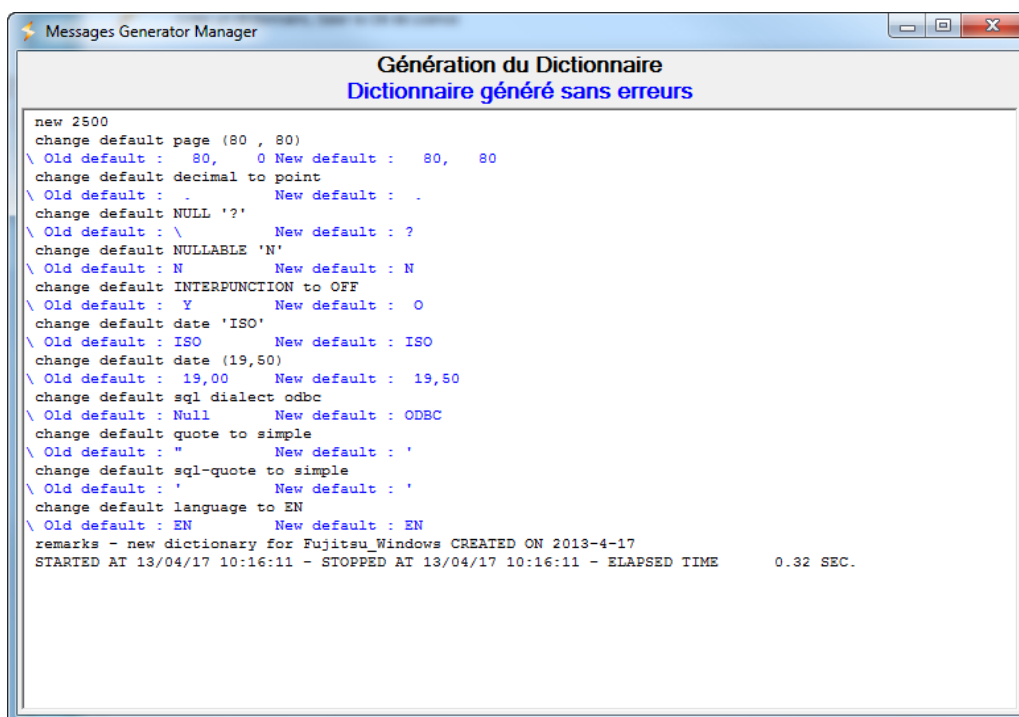
Champ	Description
Langue	Réservé pour usage ultérieur.
Importer les tables personnalisées	Cette option permet d'utiliser des tables du générateur personnalisées sauvegardées dans le dossier DCT. Initialement, ce flag prend la même valeur que le flag "Utiliser les tables personnalisées" dans INI Manager, mais l'Utilisateur peut modifier le flag si nécessaire. Si cette option est activée, les personnalisations de l'Utilisateur seront automatiquement incluses lors de la génération du Dictionnaire.

3. Créer le Fichier de Dictionnaire.

Cliquez sur le bouton *Créer* après avoir remplis tous les champs requis.

Le nouveau Fichier de Dictionnaire sera généré et il écrasera le Fichier de Dictionnaire existant, si présent.

Une fenêtre listant les anciens paramètres et les nouveaux paramètres, ainsi que tout message d'erreur s'affiche.



3.2. Saisir une Clé de licence

Cette fonction permet de saisir une nouvelle Clé de licence. Cela est obligatoire si l'ancienne Clé de licence est périmée. En fonction de votre contrat, vous pouvez commander une nouvelle Clé de licence auprès de IKAN Solutions N.V. ou la nouvelle Clé de licence vous sera envoyée automatiquement par IKAN Solutions N.V.

1. Accédez à l'écran *Créer un Dictionnaire/ Saisir la Clé de licence* en cliquant sur l'icône appropriée dans la colonne de gauche.

L'écran suivant s'affiche:

2. Complétez la section *Clé de licence*.

Il contient les champs suivants:

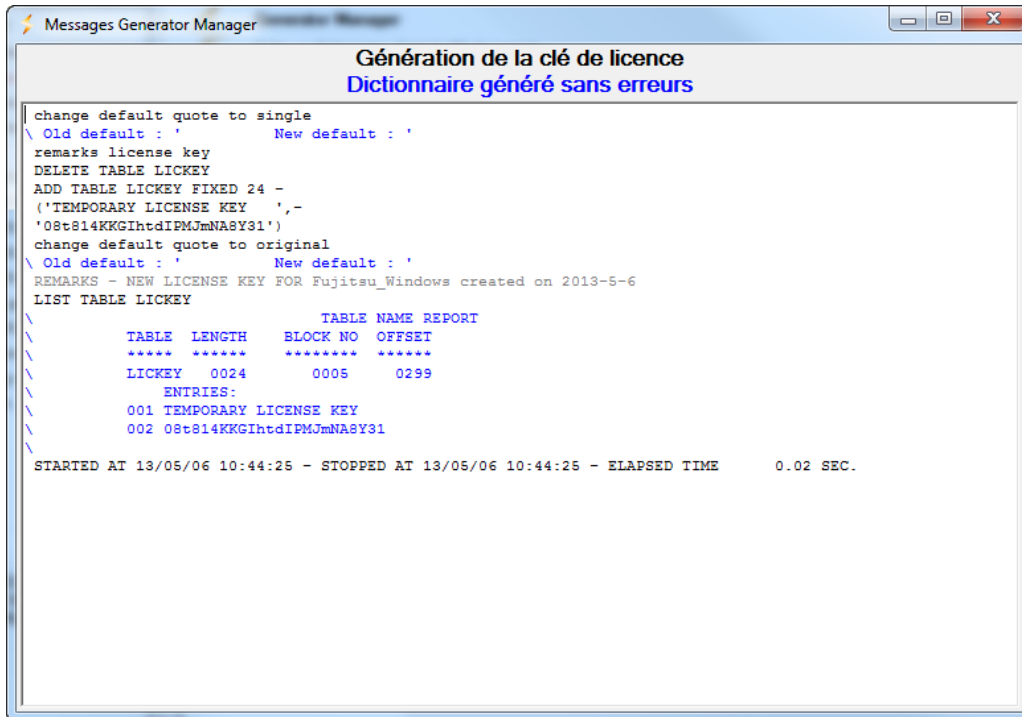
Champ	Description
Nom du client	Saisissez le nom de votre organisation. Utilisez l'orthographe exacte telle que fournie par IKAN Solutions N.V.
Clé de licence	Saisissez (ou collez) la Clé de licence, telle que fournie par IKAN Solutions N.V., dans ce Champ.

Note: Si vous utilisez plus d'un Générateur, une clé de licence doit être générée pour chaque Générateur.

3. Générer la Clé de licence

Cliquez sur le bouton *Appliquer* après avoir remplis tous les champs requis.

Des informations concernant la génération de la Clé de licence s'affichent.



The screenshot shows a window titled "Messages Generator Manager" with a subtitle "Génération de la clé de licence" and "Dictionnaire généré sans erreurs". The main content area displays the following text:

```

change default quote to single
\ Old default : '          New default : '
remarks license key
DELETE TABLE LICKEY
ADD TABLE LICKEY FIXED 24 -
('TEMPORARY LICENSE KEY ',-
'08t814KKGIhtdIPMJmN8Y31')
change default quote to original
\ Old default : '          New default : '
REMARKS - NEW LICENSE KEY FOR Fujitsu_Windows created on 2013-5-6
LIST TABLE LICKEY
\
\          TABLE NAME REPORT
\          TABLE LENGTH  BLOCK NO  OFFSET
\          *****
\          LICKEY  0024      0005     0299
\          ENTRIES:
\          001 TEMPORARY LICENSE KEY
\          002 08t814KKGIhtdIPMJmN8Y31
\
STARTED AT 13/05/06 10:44:25 - STOPPED AT 13/05/06 10:44:25 - ELAPSED TIME      0.02 SEC.

```

L'écran Options du Dictionnaire

Les Options du Dictionnaire consistent en les variables MTL définies dans la table MTLOPT. Ces options sont utilisées dans les tables techniques du Dictionnaire permettant de générer le COBOL (MGL) et le fichier de commandes (MRL) en utilisant le [Langage de modèle MetaSuite](#) (page 99).

Les Utilisateurs peuvent définir leurs propres options MTL en éditant la table MTLOPT. Vous pouvez accéder à cette table en utilisant l'option [Maintenance des tables techniques - Maintenance des tables MGL](#) (page 58). Les options MTL doivent être précédées par des lignes de commentaires commençant par ">*" et la commande SET doit se trouver immédiatement après le caractère ">" dans la deuxième colonne. Les variables alphanumériques doivent être mises entre guillemets (simples ou doubles). Les variables booléennes peuvent être 'activées' ou 'désactivées'.

1. Accédez à l'écran *Options du Dictionnaire* en cliquant sur l'icône appropriée dans la colonne de gauche.

L'écran suivant s'affiche:

Nom de l'option	Valeur de l'option	Commentaire
BLANK-RECORD-BEFORE-READ	☐ OFF	This flag can be used in order to initialize the record buffer before doing a read operation. This might lead to unexpected errors when using it on variable source files.
CHECKPOINT-CALL		Checkpoint program used for a WARM restart. CHECKPOINT-CALL contains the name of the program that saves record information of the last updated record. The default value is "MSRST@" on
CHECKPOINT-WEIGHT-CALL		Checkpoint program used for a WARM restart. CHECKPOINT-WEIGHT-CALL contains the name of the program that simulates the work of IMS program "U99CPZ". The default value is "DUMMY" on non-IBM
DATE-SEP		The character used as separator in Date Format
DATE-SEP-CHK	☐ OFF	This flag can be used in order to determine the character used as separator in Date Format. It can be set OFF or ON.
DELIMITED-CHECK-OUTER-QUOTES	☒ ON	This option is used to handle the reading of delimited source files. It can be set to the following values:

Réinitialiser Appliquer les options aux Modèles exportés Enregistrer Générer

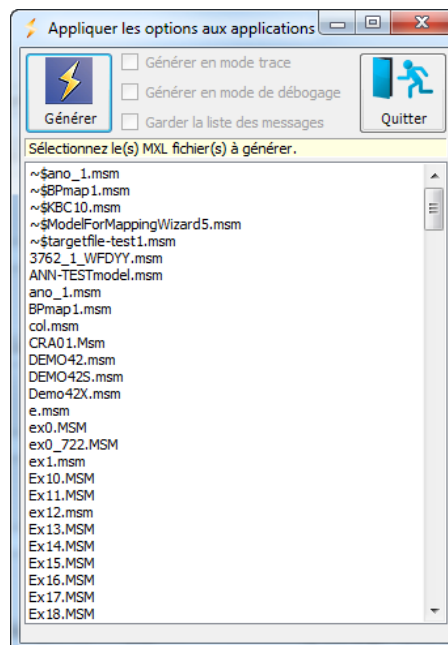
Pour chaque option disponible, les informations suivantes sont affichées:

- le *Nom de l'option*
- la *Valeur de l'option*
- *Commentaire* concernant l'usage de l'option

Seul le Champ *Valeur de l'option* peut être modifié. En fonction de l'option vous pouvez saisir une valeur numérique, une valeur alphanumérique ou modifier le statut d'activation.

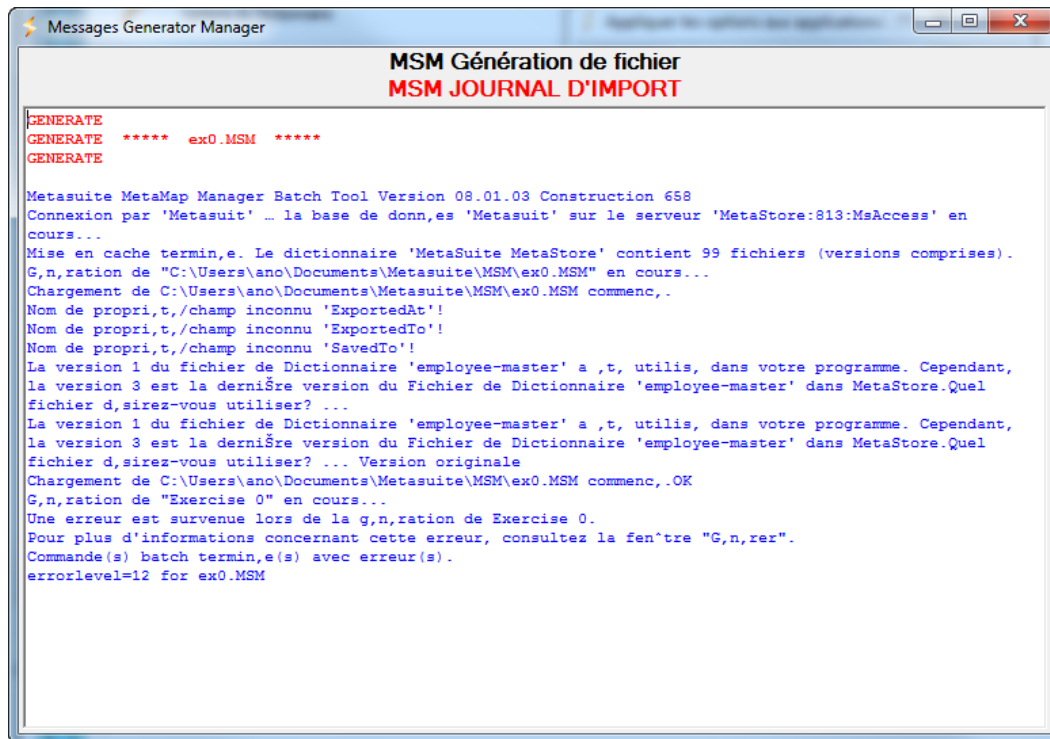
2. Modifiez les paramètres des options MTL requises.
3. Une fois les modifications requises sont faites, vous pouvez exécuter une des actions suivantes:
 - *Appliquer les options aux Modèles exportés*

La fenêtre suivante s'affiche:



Sélectionnez une ou plusieurs applications et cliquez sur le bouton *Générer*.

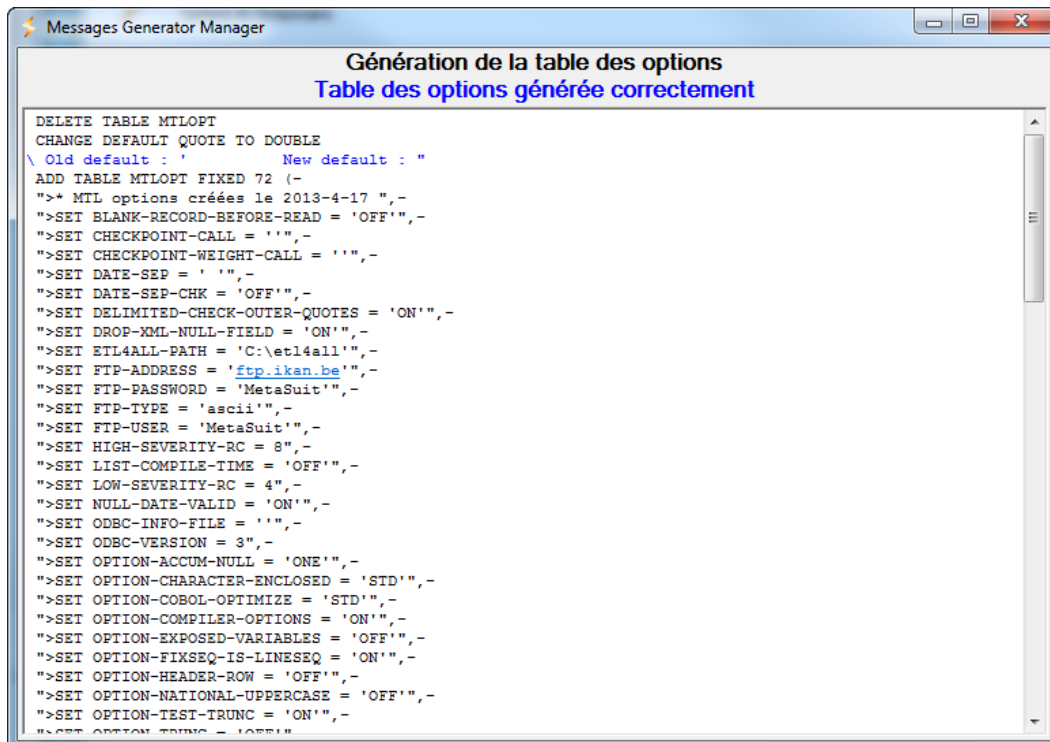
La fenêtre *Génération de fichier MIL* s'affiche (si vous avez coché la case *Garder la liste des messages*):



- *Générer*

La génération des paramètres d'une option MTL implique leur implémentation immédiate dans le Dictionnaire du Générateur.

La fenêtre suivante s'affiche:



Écran de génération - Aperçu

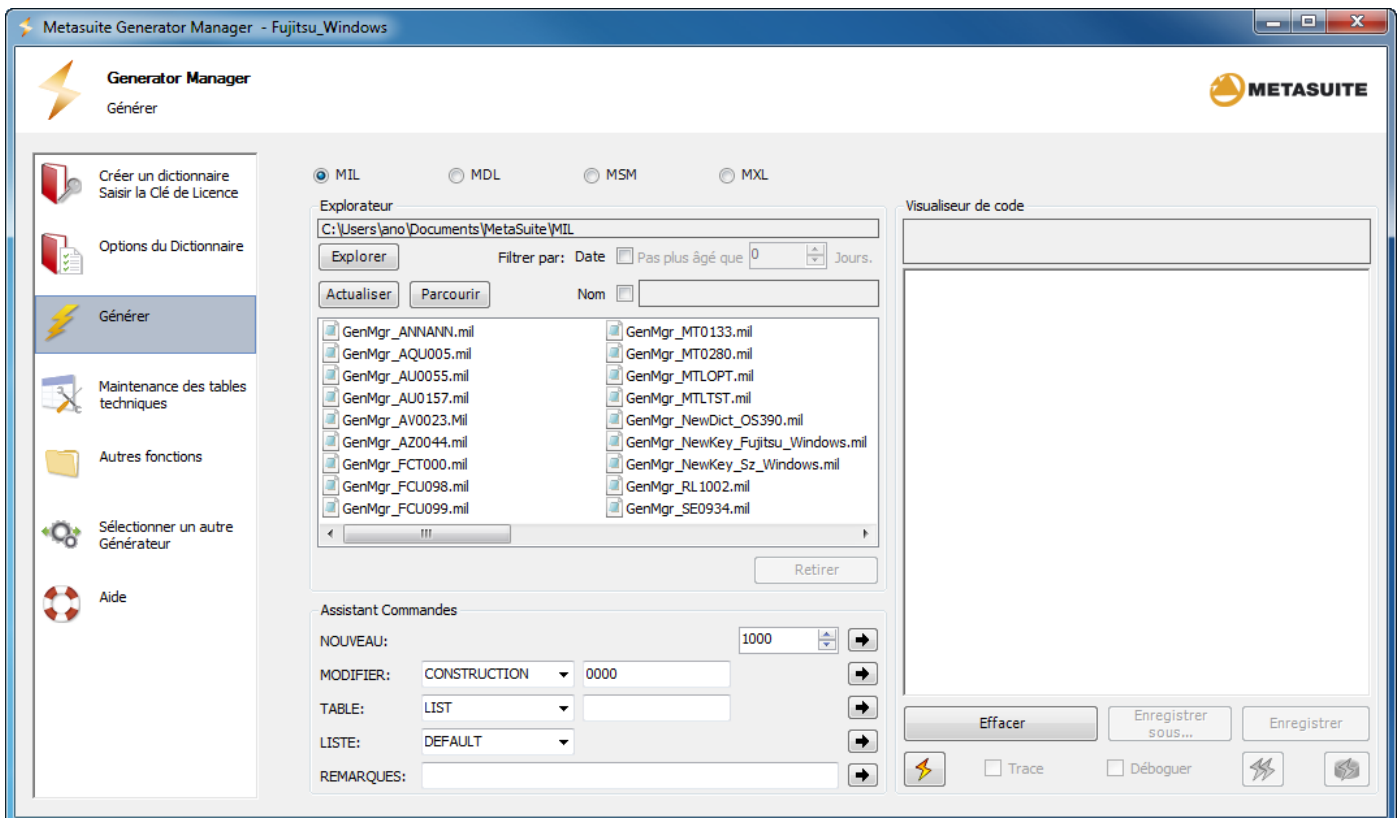
L'écran *Générer* permet de générer les différents types de fichier: MIL, MDL, MSM et MXL.

Pour plus d'informations, se référer aux chapitres spécifiques:

- [Générer - Implémenter les instructions MIL](#) (page 18)
- [Générer les fichiers MDL](#) (page 37)
- [Générer les fichiers MSM](#) (page 47)
- [Générer les fichiers MXL](#) (page 49)

1. Accédez à l'écran *Générer* en cliquant sur l'icône appropriée dans la colonne de gauche.

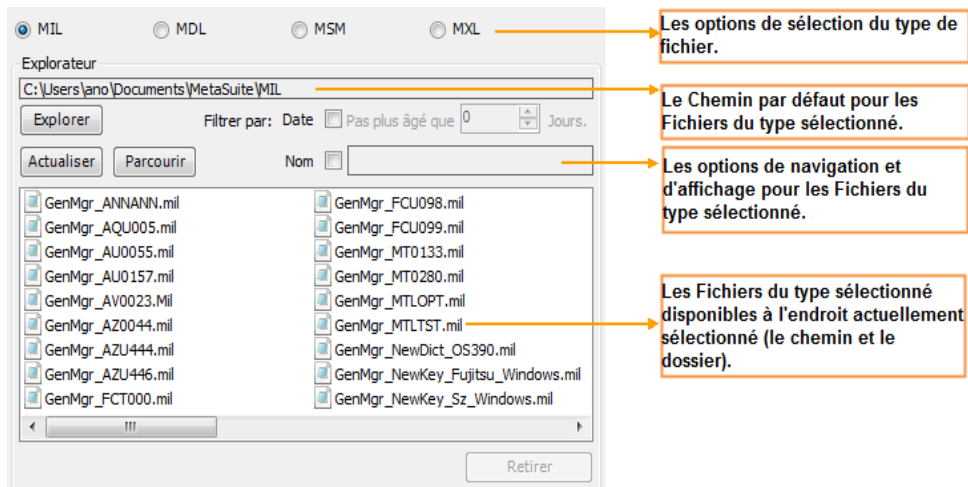
L'écran suivant s'affiche:



Les éléments en commun sont affichés en haut à gauche:

- Types de fichier
- Options de l'Explorateur
- Assistant Commandes
- Éditeur de code

- Actions possibles



2. Sélectionnez le type de fichier requis.

Les types suivants sont disponibles:

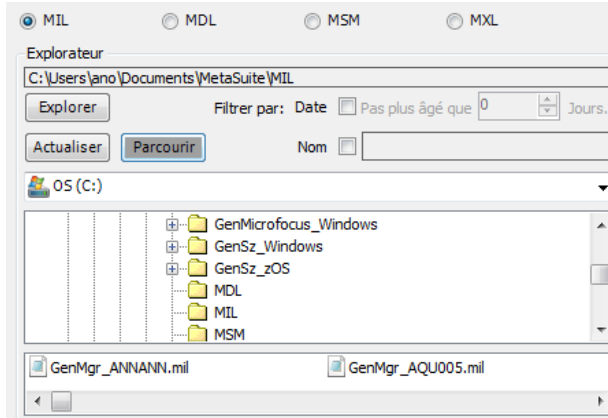
- [MIL](#) (page 18)
- [MDL](#) (page 37)
- [MSM](#) (page 47)
- [MXL](#) (page 49)

3. Vérifiez le chemin.

Le chemin indiqué est le chemin par défaut pour le type de fichier sélectionné.

4. Si vous ne trouvez pas immédiatement le fichier requis dans la liste des fichiers disponibles, vous pouvez utiliser les options de l'*Explorateur*:

Option	Description
Explorer	Sélectionnez cette option pour ouvrir l'Explorateur Windows. Accédez à un autre dossier si nécessaire. Vous pouvez également copier, modifier ou supprimer les fichiers disponibles.
Actualiser	Cette option permet d'actualiser la liste des Fichiers disponibles dans le dossier sélectionné. Il peut être utile de sélectionner cette option si le fichier attendu n'est pas affiché.

Option	Description
Parcourir	Cette option permet de sélectionner un autre dossier contenant les fichiers du type sélectionné: • Cliquez sur le bouton <i>Parcourir</i>. La structure hiérarchique du dossier s'affiche:  • Accédez au dossier contenant les fichiers requis. • Cliquez à nouveau sur le bouton <i>Parcourir</i> pour cacher la hiérarchie.
Filtrer par date	Cochez cette case si vous voulez définir un filtre sur la base d'une date. Saisissez le nombre de jours dans le champ <i>Pas plus âgé que</i> et appuyez sur ENTER. Par exemple: si vous saisissez 30, le système recherchera tous les fichiers disponibles pas plus âgés que 30 jours.
Filtrer par nom	Cochez cette case si vous voulez définir un filtre sur la base du nom du Fichier. Saisissez une chaîne de caractères. La liste des fichiers disponibles sera limitée aux fichiers dont le nom contient la chaîne de caractères spécifiée. Décochez cette case pour afficher à nouveau tous les fichiers disponibles.

Générer - Implémenter les instructions MIL

MIL signifie MetaSuite Installation Language (Langage d'installation MetaSuite). Cette option permet d'exécuter une liste d'instructions MIL. Vous pouvez établir la liste de trois façons:

1. [Copier des Fichiers MIL existants](#) (page 20)
2. [Ajouter des instructions MIL en utilisant l'Assistant Commandes](#) (page 21)
3. Saisir les instructions MIL manuellement

Lors de l'exécution des instructions MIL, les paramètres référencés sont mis à jour dans le Fichier de Dictionnaire. Vous ne devez pas recréer le Dictionnaire entier. Les paramètres non modifiés par les instructions MIL exécutées resteront identiques.

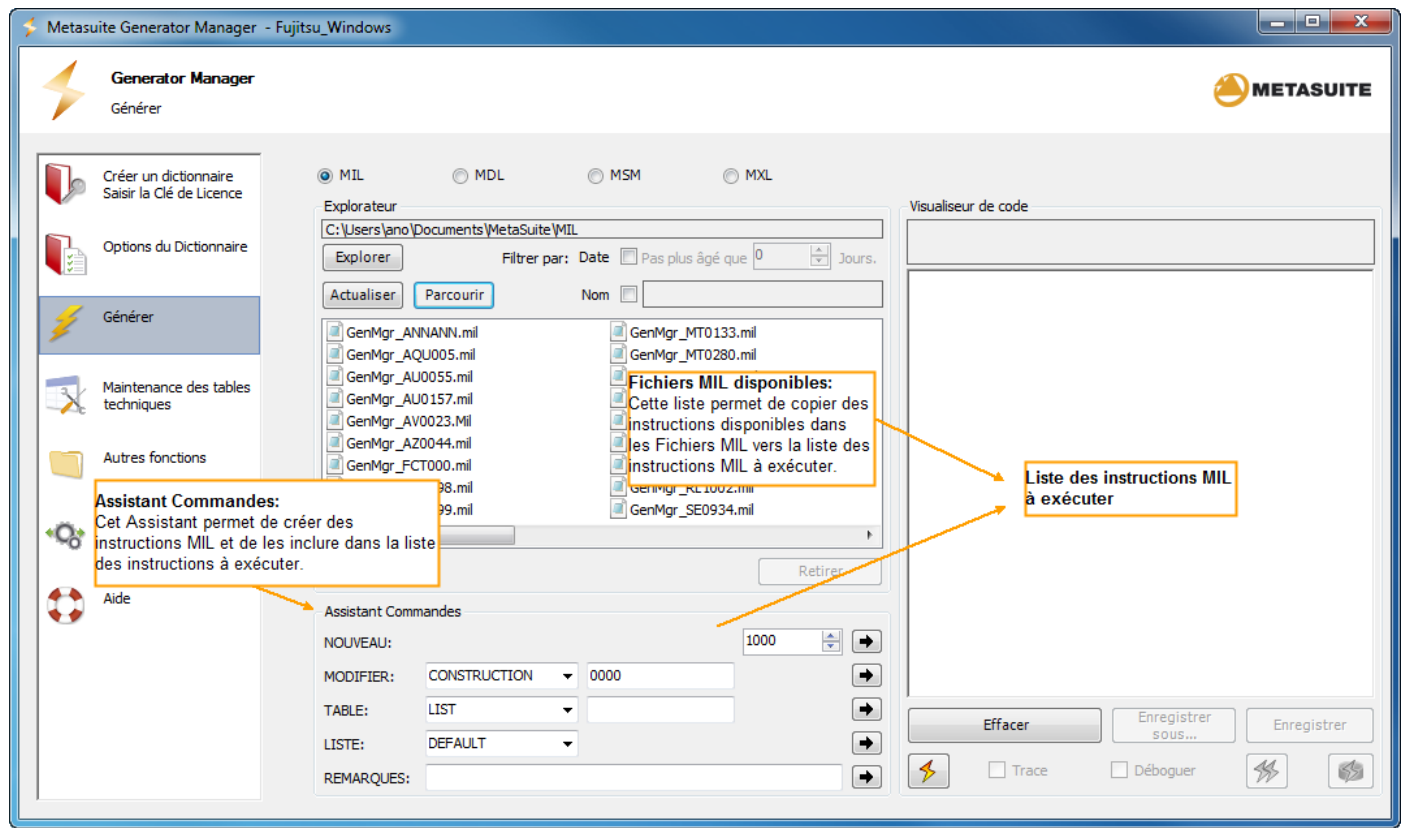
Ce chapitre contient également les sections de référence suivantes:

- [Référence des options CHANGE](#) (page 24)
- [Options de TABLE - Référence](#) (page 33)

6.1. L'écran "Générer pour MIL"

1. Cliquez sur le bouton *Générer* dans la colonne de gauche.
2. Sélectionnez l'option de format *MIL*.

L'écran suivant s'affiche:

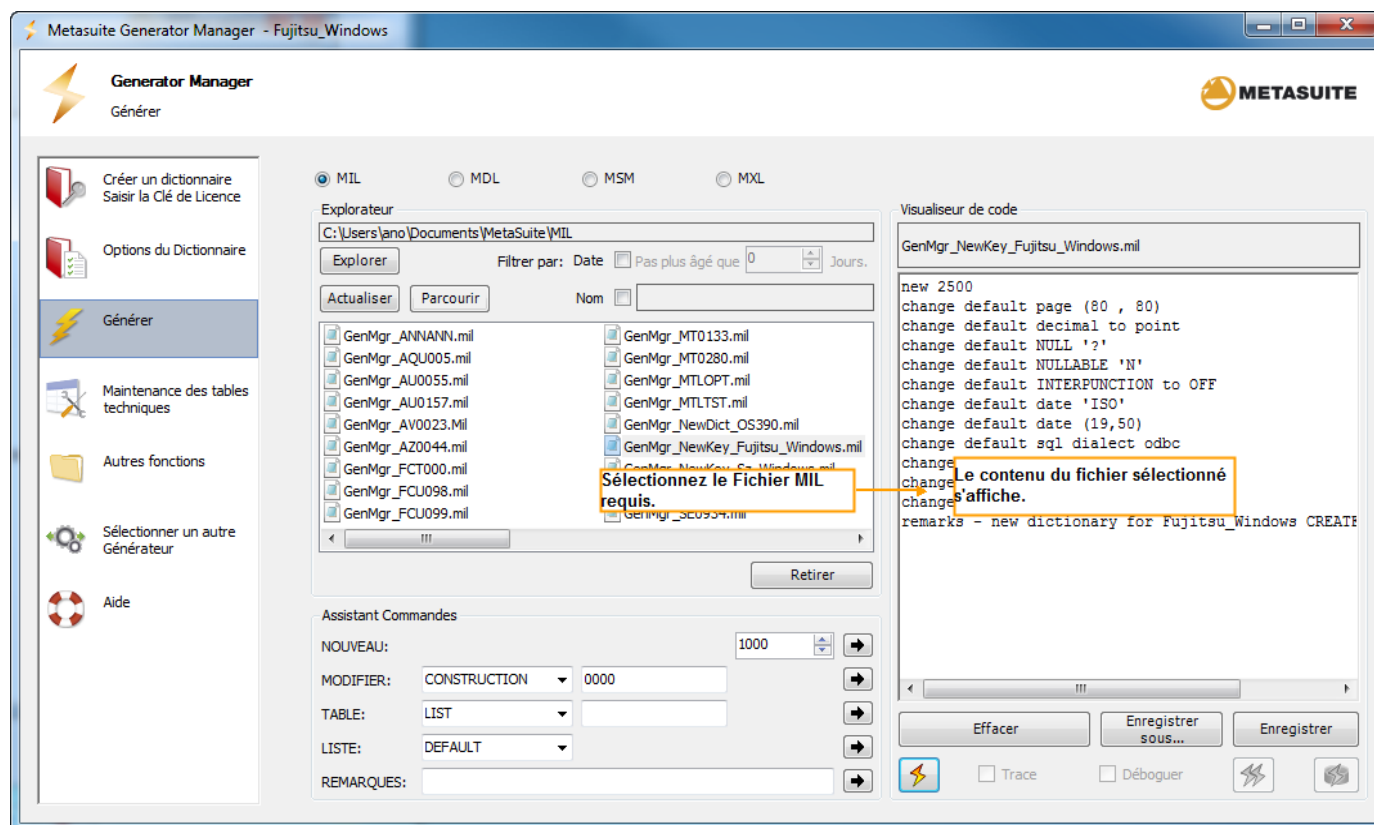


La fenêtre des Instructions MIL est composée de trois sections:

Section	Description
Liste des Fichiers MIL disponibles	La liste s'affiche en haut à gauche. Sélectionnez un Fichier pour ajoute son contenu à la liste des instructions MIL à exécuter affichée à droite. Voir Copier des Fichiers MIL existants on page 20.
Assistant Commandes	Utilisez l'Assistant Commandes pour créer des instructions MIL manuellement pour les inclure dans la liste des instructions MIL à exécuter. Voir Ajouter des instructions MIL en utilisant l'Assistant Commandes on page 21.
Liste des instructions MIL à exécuter	La liste contient les instructions MIL ajoutées à partir d'un Fichier MIL et/ou à partir de l'Assistant Commandes. Vous pouvez manuellement modifier les instructions listées ou ajouter des instructions additionnelles à la liste.

6.2. Copier des Fichiers MIL existants

1. Accédez à l'écran des instructions MIL.
Voir [L'écran "Générer pour MIL"](#) on page 19.
2. Sélectionnez le Fichier MIL requis à partir de la liste des Fichiers MIL disponibles.
Son contenu est affiché immédiatement dans la liste des instructions MIL à implémenter:

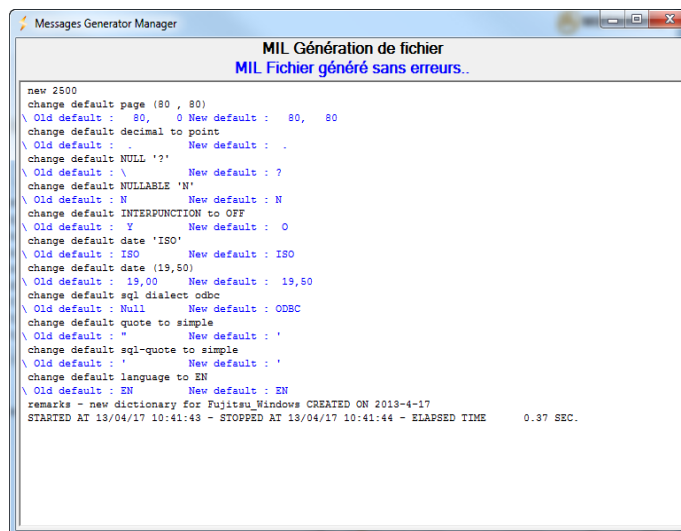


Note: Si vous sélectionnez un autre Fichier MIL, le contenu de ce fichier remplacera les instructions actuellement affichées. Vous pouvez supprimer un Fichier MIL, en le sélectionnant à partir de la liste des Fichiers MIL disponibles et en cliquant ensuite sur le bouton *Supprimer* en bas de l'écran.

3. Si nécessaires, modifiez les commandes MIL affichées.
Vous pouvez également utiliser l'Assistant Commandes pour insérer les commandes. Voir [Ajouter des instructions MIL en utilisant l'Assistant Commandes](#) on page 21.
4. Une fois les modifications requises sont faites, vous pouvez exécuter une des actions suivantes:

Action	Description
Effacer	Efface le contenu de la liste pour pouvoir redémarrer.
Enregistrer sous	Sauvegarde le nouveau Chemin et le nouveau Nom de Fichier dans un nouveau Fichier MIL. Saisissez le nouveau Chemin et le nouveau Nom de fichier et cliquez sur le bouton OK.

Action	Description
Enregistrer	Sauvegarder les commandes MIL affichées (modifiées) dans le Fichier MIL existant.
	Cliquez sur le bouton Générer si vous voulez implémenter les commandes MIL affichées. Les commandes MIL seront implémentées pour que les nouveaux paramètres deviennent valides dans le Fichier de Dictionnaire. La fenêtre suivante confirme cette action:



6.3. Ajouter des instructions MIL en utilisant l'Assistant Commandes

1. Accédez à l'écran des instructions MIL.
Voir [L'écran "Générer pour MIL"](#) on page 19.

Note: Si nécessaire, sélectionnez un Fichier MIL à partir de la liste de Fichiers MIL disponibles. Voir [Copier des Fichiers MIL existants](#) on page 20.

L'Assistant Commandes se situe dans le coin gauche en bas.

The 'Assistant Commandes' dialog box is shown with the following fields and controls:

- NOUVEAU:** A text input field containing '1000' and a right-pointing arrow button.
- MODIFIER:** A dropdown menu set to 'CONSTRUCTION' and a text input field containing '0000', with a right-pointing arrow button.
- TABLE:** A dropdown menu set to 'LIST' and a text input field, with a right-pointing arrow button.
- LISTE:** A dropdown menu set to 'DEFAULT' and a text input field, with a right-pointing arrow button.
- REMARQUES:** A text input field with a right-pointing arrow button.

Chaque ligne représente une catégorie d'instructions MIL.

2. Sélectionnez les instructions requises et cliquez sur le bouton  pour les ajouter à la liste des instructions MIL à droite.

Les catégories suivantes sont disponibles:

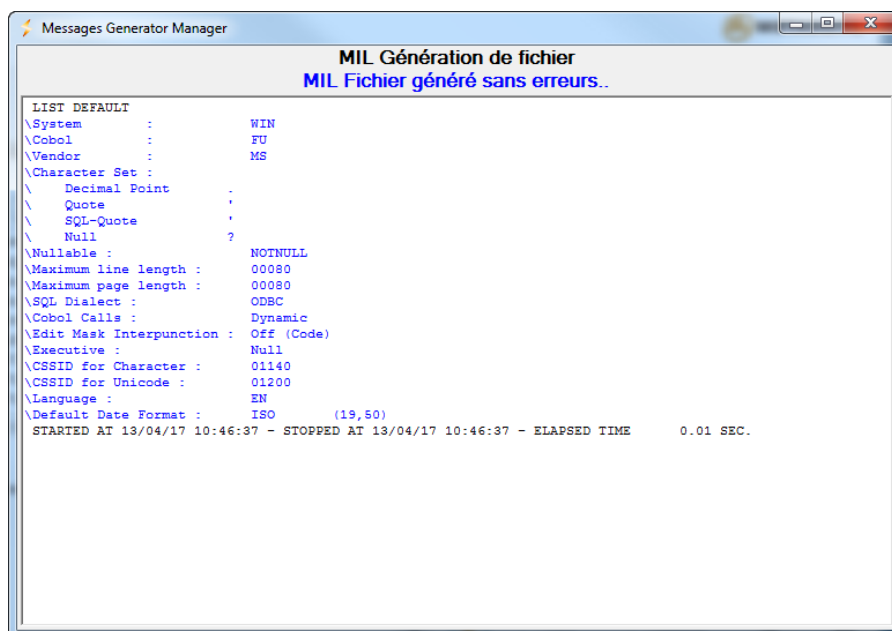
Catégorie	Description
NOUVEAU	Cette Catégorie permet d'insérer une commande <i>NEW</i> dans la liste des commandes MIL à implémenter. La commande <i>NEW</i> crée une nouvelle bibliothèque de Générateur. Elle doit être la première commande si vous effectuez un chargement initial d'une nouvelle bibliothèque du Générateur. Dans le Champ numérique à droite: 1. Saisissez le nombre à quatre chiffres représentant le nombre de blocs à initialiser pour la nouvelle bibliothèque du Générateur. 2. Cliquez sur la flèche.
MODIFIER	Pour ajouter une instruction <i>CHANGE</i>: 1. Sélectionnez l'élément à changer à partir de la liste déroulante. 2. Sélectionnez la valeur de l'élément 3. Cliquez sur la flèche. Les options suivantes sont disponibles: • CARACTÈRE-CCSID (page 32) • CONSTRUCTION (page 25) • DATE BASCULE SIÈCLE (page 26) • DÉCIMALE (page 26) • DYNAMIQUE (page 31) • EXEC (page 27) • FORMAT DE DATE (page 25) • GUILLEMETS (page 29) • INTER-PONCTUATION (page 27) • LANGAGE D'EXPORT (page 32) • NULL (page 28) • NULLABLE (page 28) • PAGE (page 29) • SQL (page 30) • SQL-QUOTE (page 31) • UNICODE-CCSID (page 32) Pour des informations plus détaillées, se référer aux sections indiquées.
TABLE	Pour ajouter une instruction <i>TABLE</i>: 1. Sélectionnez l'action de table requise à partir de la liste déroulante 2. Saisissez le nom de la table à afficher. 3. Cliquez sur la flèche. Les actions suivantes sont disponibles: • LIST (page 33) • ADD (page 34) • COPY (page 35) • DELETE (page 35) • IMPORT (page 36) • EXPORT (page 35) Pour des informations plus détaillées, se référer aux sections indiquées.

Catégorie	Description
-----------	-------------

LIST

Les options suivantes sont disponibles:

- **DEFAULT**
Sélectionnez DEFAULT pour afficher la liste de toutes les valeurs par défaut s'appliquant à la bibliothèque du Générateur.
Le type d'information suivant s'affichera:



- **VERSION**

Sélectionnez VERSION pour afficher la version et le numéro de construction du Générateur et de la bibliothèque du Générateur courants. Le résultat de cette commande contient également les exigences du Générateur et de la bibliothèque du Générateur: une construction spécifique d'un Générateur correspond à un numéro de construction spécifique du Dictionnaire, et vice versa.

Le type d'information suivant s'affichera:



REMARQUES

Pour ajouter une commande REMARKS à la liste des instructions MIL:

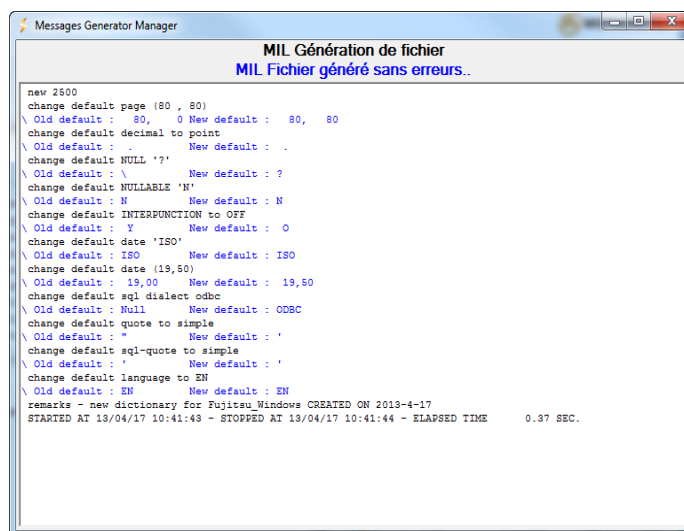
1. Dans le champ de texte, saisissez les remarques que vous voulez ajouter
2. Cliquez sur la flèche.

3. Une fois les instructions MIL ajoutées, vous pouvez exécuter une des actions suivantes:

Action	Description
Effacer	Efface le contenu de la liste pour pouvoir redémarrer.
Enregistrer sous	Sauvegarde le nouveau Chemin et le nouveau Nom de Fichier dans un nouveau Fichier MIL. Saisissez le nouveau Chemin et le nouveau Nom de fichier et cliquez sur le bouton OK.
Enregistrer	Sauvegarder les commandes MIL affichées (modifiées) dans le Fichier MIL existant.



Cliquez sur le bouton **Générer** si vous voulez implémenter les commandes MIL affichées. Les commandes MIL seront implémentées pour que les nouveaux paramètres deviennent valides dans le Fichier de Dictionnaire.
La fenêtre suivante confirme cette action:



6.4. Référence des options CHANGE

Cette section contient des informations de référence concernant les options de CHANGE suivantes:

- [CONSTRUCTION](#) (page 25)
- [FORMAT DE DATE](#) (page 25)
- [DATE BASCULE SIÈCLE](#) (page 26)
- [DÉCIMALE](#) (page 26)
- [EXEC](#) (page 27)
- [INTER-PONCTUATION](#) (page 27)
- [NULL](#) (page 28)
- [NULLABLE](#) (page 28)
- [PAGE](#) (page 29)
- [GUILLEMETS](#) (page 29)
- [SQL](#) (page 30)
- [SQL-QUOTE](#) (page 31)
- [DYNAMIQUE](#) (page 31)

- [LANGAGE D'EXPORT](#) (page 32) (réservé pour usage ultérieur)
- [UNICODE-CCSID](#) (page 32)
- [CARACTÈRE-CCSID](#) (page 32)

Pour une description détaillée de la procédure, se référer aux sections suivantes:

- [Copier des Fichiers MIL existants](#) (page 20)
- [Ajouter des instructions MIL en utilisant l'Assistant Commandes](#) (page 21)

CONSTRUCTION

La commande *CHANGE DEFAULT BUILD* permet de modifier le numéro de construction du Dictionnaire après avoir appliqué un "patch". Grâce à cette commande, la génération d'un nouveau Dictionnaire peut être évitée.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *CONSTRUCTION* à partir de la liste déroulante.
2. Saisissez le numéro de construction du Dictionnaire dans le champ numérique à droite.
Ce numéro de construction est obligatoire et doit être un numéro entre 00 et 99.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT BUILD* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

FORMAT DE DATE

La commande *CHANGE DEFAULT DATE* permet de définir le format de date à utiliser pour la manipulation de dates dans un environnement RDBMS.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *FORMAT DE DATE* à partir de la liste déroulante.
Une nouvelle liste déroulante contenant les options de format de date disponibles s'affiche à droite.
2. Sélectionnez l'option du format de date requise.

Option	Format
ISO	YYYY-MM-DD
EUR	DD.MM.YYYY
JIS	YYYY-MM-DD
USA	MM/DD/YYYY

Note: Dans un environnement RDBMS, le format de date des données Source doit être établi à ISO. Ceci peut être effectué en utilisant une option de pré-compilation ou de compilation. Le format de date de données Source à partir d'un Fichier plat est défini dans le Dictionnaire MetaStore. Le délimiteur entre les dates de type caractère n'est pas important.

3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT DATE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

DATE BASCULE SIÈCLE

La commande *CHANGE DATE BREAK YEAR* permet de déterminer quelles années font partie du vingtième ou du vingt-et-unième siècle si le format de date n'inclut pas d'indication de siècle, comme par exemple dans le format de date *YYMMDD*.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *DATE BASCULE SIÈCLE* à partir de la liste déroulante.
2. Spécifiez le siècle et l'année de bascule pour les indications de date pour lesquelles les années sont spécifiées comme un numéro à 2 chiffres.

La combinaison de "année de bascule" et "siècle de bascule" est utilisée pour convertir des années représentées par deux chiffres en années représentées par quatre chiffres. La combinaison des deux paramètres forme l'année la plus ancienne (la plus basse) depuis laquelle une année représentée par deux chiffres peut être convertie.

- Dans le premier champ, saisissez le siècle le plus bas.
- Dans le deuxième champ, saisissez le point de bascule de l'année.

Par exemple:

Si vous spécifiez (19 , 60) comme siècle de bascule et année de bascule, 1960 sera l'année la plus ancienne possible.

- Les années indiquées par les chiffres 00 à 59 sont interprétées comme les années 2000 à 2059.
- Les années indiquées par les chiffres 60 à 99 sont interprétées comme les années 1960 à 1999.

3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DATE BREAK YEAR* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

DÉCIMALE

La commande *CHANGE DEFAULT DECIMAL* permet de sélectionner le Point décimal requis. Vous pouvez choisir entre *Virgule* et *Point*.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *DÉCIMALE* à partir de la liste déroulante.
2. Sélectionnez *Point* ou *Virgule*.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT DECIMAL* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

EXEC

La commande *CHANGE DEFAULT EXEC* permet de sélectionner le mode d'exécution requis.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *EXEC* à partir de la liste déroulante.
Une deuxième liste déroulante s'affiche à droite.
2. Sélectionnez le mode d'exécution requis.

Mode d'exécution	Explication
(aucun)	Sélectionnez <i>(aucun)</i> dans toutes les autres situations.
IMS	Cette option permet le redémarrage sous IMS/DC.
REDÉMARRABLE	Pour les autres environnements, cette option permet un redémarrage.

3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT EXEC* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

INTER-PONCTUATION

La commande *CHANGE DEFAULT INTERPUNCTION* est utilisée pour spécifier le masque d'édition par défaut pour les valeurs numériques.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *INTER-PONCTUATION* à partir de la liste déroulante.
Une deuxième liste déroulante s'affiche à droite.

2. Sélectionnez l'option de ponctuation requise.

Ponctuation	Explication
OFF	Chaque champ numérique doit être traité comme un code, c.-à-d. sans compression de zéros et sans inter-ponctuation.
NON	Le masque d'édition par défaut pour les Champs numériques (non codifiés) doit spécifier la suppression des zéros mais pas d'inter-ponctuation. Notez que les zéros ne seront pas supprimés dans les codes numériques et que l'inter-ponctuation ne sera pas ajoutée.
OUI	Le masque d'édition par défaut pour les Champs numériques (non codifiés) doit spécifier la suppression des zéros et l'ajout de l'inter-ponctuation et que l'inter-ponctuation ne sera pas ajoutée. Notez que les zéros ne seront pas supprimés dans les codes numériques.

3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT INTERPUNCTION* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

NULL

La commande *CHANGE DEFAULT NULL* permet de définir le caractère simple qui sera utilisé pour indiquer la valeur Null dans un Champs d'un Fichier séquentiel.

Le Caractère Null à utiliser pour le stockage INBOUND et OUTBOUND NULL dans un Champ.

Dans le cas de INBOUND NULL, vous devez choisir le Caractère Null avec précaution vu que l'apparition du Caractère Null à la première position du champ détermine si le champ contient une valeur pour Null.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *NULL* à partir de la liste déroulante.
Un champ de texte contenant le caractère ? s'affiche à droite.
2. Remplacez ce caractère par le caractère pour Null requis.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT NULL* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

NULLABLE

La commande *CHANGE DEFAULT NULLABLE* permet de définir l'option *NULLABLE* par défaut pour les Champs.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *NULLABLE* à partir de la liste déroulante.
Une nouvelle liste déroulante s'affiche à droite.
2. Sélectionnez l'option requise.

Option	Explication
N	Tous les Champs pour lesquels aucun paramètre DBNAME spécifique n'est défini seront traités comme étant NOT-NULLABLE (c.-à-d. DBNAME 'NOTNULL'). Le Caractère Null ne sera pas interprété pour ces Champs.
I	Tous les Champs pour lesquelles aucun paramètre DBNAME spécifique n'est défini seront traités comme étant INBOUND-NULLABLE (c.-à-d. DBNAME 'INNULL').

3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT NULLABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

PAGE

La commande *CHANGE DEFAULT PAGE* permet de définir la longueur de page par défaut (en nombre de lignes) et la largeur de page (en nombre de caractères) qui sera utilisée dans les Rapports Cible si aucun paramètre n'est spécifié pour la page par le Modèle MetaMap.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *PAGE* à partir de la liste déroulante.
Deux champs de texte s'affichent à droite.
2. Saisissez une nouvelle valeur par défaut pour la longueur de page dans le premier champ de texte.
3. Saisissez une nouvelle valeur par défaut pour la Largeur de page dans le deuxième champ de texte.
4. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT PAGE* dans la fenêtre des commandes.
5. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

GUILLEMETS

La commande *CHANGE DEFAULT QUOTE* permet de définir les guillemets requis. Vous pouvez choisir entre *Simple* et *Double*.

Ce caractère de guillemet est utilisé pour entourer les phrases de table dans les Fichiers MIL et pour entourer les littéraux COBOL dans les Fichiers MGL.

La principale utilisation de la commande *CHANGE DEFAULT QUOTE* est de modifier temporairement les guillemets par défaut en guillemets doubles, avant d'ajouter ou de modifier une ou plusieurs définitions de table. Si vous modifiez les guillemets par défaut avant de saisir la commande *ADD TABLE*, vous pouvez embarquer des guillemets simples dans le texte de toute entrée de table à ajouter ou à remplacer. Après avoir saisi toutes les commandes de table, vous devez établir la valeur par défaut à nouveau à des guillemets simples en codifiant une deuxième commande *CHANGE DEFAULT QUOTE*.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *GUILLEMETS* à partir de la liste déroulante.
Vous pouvez choisir entre des guillemets simples et doubles. Deux boutons d'option s'affichent à droite.
2. Cochez *Simple* ou *Double*.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT QUOTE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

SQL

La commande *CHANGE DEFAULT SQL DIALECT* permet de définir le dialecte SQL à utiliser lors de la génération de SQL imbriqué et quand aucun dialecte SQL spécifique n'est défini dans le Modèle MetaMap.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *SQL* à partir de la liste déroulante.

Une nouvelle liste déroulante s'affiche à droite, contenant les options suivantes:

- DB2 pour OS/400
- DB2 pour z/OS
- DB2 LUW
- DB2/2
- DB2/VSE
- Générateur par défaut
- Informix
- Ingres
- MySQL
- ODBC
- Oracle
- Oracle/RDB
- SESAM
- SQL Server
- Sybase
- Teradata

2. Sélectionnez le dialecte SQL requis.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT SQL DIALECT* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

SQL-QUOTE

La commande *CHANGE DEFAULT SQL-QUOTE* modifie le paramètre des guillemets pour les commandes SQL dans les programmes COBOL générés.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *SQL-QUOTE* à partir de la liste déroulante.
Vous pouvez choisir entre des guillemets simples et doubles.
2. Cochez *Simple* ou *Double*.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT SQL-QUOTE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

DYNAMIQUE

La commande *CHANGE DEFAULT DYNAMIC* décrit comment le programme MetaSuite appelle les modules d'exécution.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *DYNAMIQUE* à partir de la liste déroulante.
Les valeurs possibles sont:
 - I = appels COBOL internes (les modules d'exécution font partie de la source COBOL générée)
 - Y = appels COBOL dynamiques
 - N = appels COBOL statiques
2. Sélectionnez l'option requise.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT DYNAMIC* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

LANGAGE D'EXPORT

Le nom est une abréviation à deux caractères de la langue des messages du Générateur.

Actuellement, les langues anglais (EN) et français (FR) sont supportées.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *LANGAGE D'EXPORT* à partir de la liste déroulante.
2. Saisissez l'abréviation de langue requise. EN ou FR.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT LANGUAGE TO* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

UNICODE-CCSID

CCSID est une abréviation utilisée par IBM signifiant "Coded Character Set Identifier". Il s'agit d'un chiffre de 16 octets représentant une codification spécifique d'une page de code spécifique. Par langue, pays ou région, un jeu de caractères spécifique est utilisé pour définir la "Page de codification" régionale.

Les jeux de caractères Unicode auraient dû échapper à ce type de comportement spécifique au langage, mais cela n'est pas le cas. Les premiers 256 caractères du jeu de caractères Unicode dépendent du CCSID.

La commande *CHANGE DEFAULT UNICODE-CCSID* définit le CCSID standard pour les jeux de caractères à double octets.

1. Sur la ligne *MODIFIER* de l'Assistant Commandes, sélectionnez *UNICODE-CCSID* à partir de la liste déroulante.
2. Saisissez la valeur requise.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT UNICODE-CCSID* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

CARACTÈRE-CCSID

CCSID est une abréviation utilisée par IBM signifiant "Coded Character Set Identifier". Il s'agit d'un chiffre de 16 octets représentant une codification spécifique d'une page de code spécifique. Par langue, pays ou région, un jeu de caractères spécifique est utilisé pour définir la "Page de codification" régionale.

La commande *CHANGE DEFAULT UNICODE-CCSID* définit le CCSID standard pour les jeux de caractères à un octet.

1. Sur la ligne *CHANGE* de l'Assistant Commandes, sélectionnez *CARACTÈRE-CCSID* à partir de la liste déroulante.
2. Saisissez la valeur requise.
3. Cliquez sur la flèche à droite pour ajouter la commande *CHANGE DEFAULT CHARACTER-CCSID* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

6.5. Options de TABLE - Référence

Cette section contient des informations de référence concernant les options de TABLE suivantes:

- [LIST](#) (page 33)
- [ADD](#) (page 34)
- [COPY](#) (page 35)
- [DELETE](#) (page 35)
- [EXPORT](#) (page 35)
- [IMPORT](#) (page 36)

LIST

La commande *LIST TABLE* permet de lister une table sélectionnée ou toutes les tables disponibles dans la bibliothèque du Générateur.

C'est commande est particulièrement utile pour produire une copie papier de la liste de chaque table de la bibliothèque du Générateur avant de la modifier ou de la supprimer, pour si jamais il s'avère nécessaire de restaurer la table à son état d'origine.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *LIST* à partir de la liste déroulante.
2. Saisissez le nom de la table à afficher dans le champ de texte à droite.
Si vous laissez ce Champ vide ou si vous saisissez *TOUT* dans ce champs, toutes les tables seront listées.
3. Cliquez sur la flèche à droite pour ajouter la commande *LIST TABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

ADD

La commande *ADD TABLE* permet d'ajouter une table à la bibliothèque du Générateur.

Il existe quatre types de base pour les tables de bibliothèque:

- Tables Système
Ces tables sont utilisées pour les mots réservés, les messages d'erreur et les tables de paramètres qui dépendent du Système.
- Tables de code COBOL
Ces tables sont les blocs de construction de la Source COBOL.
- Tables de Code-Contrôle
Ces tables sont définies en conjonction avec l'option *CODE-CONTROL* dans MetaMap. L'Utilisateur peut ajouter des Tables de Code-Contrôle définies par lui-même au Dictionnaire.
- Tables de Code de Fichier
Ces tables contiennent une chaîne de 32 caractères. Chaque caractère représente une option. Les Tables de Contrôle des Fichiers sont en train de devenir obsolètes. Elles sont remplacées par les Tables de Code-Contrôle.

Cette option est surtout utilisée pour définir les Tables de Code-Contrôle et leurs Tables de Code Prototype associées.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *ADD* à partir de la liste déroulante.
2. Dans le premier champ de texte au centre, saisissez le nom de la table en cours de définition dans la bibliothèque.
Le nom
 - doit être unique dans la bibliothèque,
 - peut aller jusqu'à six caractères de longueur,
 - doit commencer par un caractère alphabétique;
 - peut contenir les caractères A-Z, 0-9 et les traits d'union imbriqués.
3. Dans le deuxième champ de texte à droite, saisissez la longueur maximale, en nombre de caractères, de toute entrée dans la table.
Toute entrée plus petite que la valeur définie sera remplie d'espaces du côté droit. La taille d'entrée maximale ne peut pas dépasser 72 pour les Tables de Code-Contrôle COBOL (MGL). Pour les autres tables, la valeur maximale est 99.
4. Cliquez sur la flèche à droite pour ajouter la commande dans la fenêtre des commandes.
5. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

COPY

La commande *COPY TABLE* est utilisée pour copier le contenu de la table de la bibliothèque du Générateur sélectionnée vers le Fichier de sortie *mscopy.mil*, qui sera sauvegardé dans le dossier suivant:

<Installation>\GENXXX\TMP

où XXX indique le Générateur sélectionné.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *COPY* à partir de la liste déroulante.
2. Dans le champ de texte à droite, saisissez le nom de la table à copier vers le Fichier *mscopy.mil*.
3. Cliquez sur la flèche à droite pour ajouter la commande *COPY TABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

DELETE

La commande *DELETE TABLE* est utilisée pour supprimer une table de la bibliothèque du Générateur.

Soyez prudent à ne pas supprimer de tables système de la bibliothèque, car cela pourrait rendre le système inutilisable. Pour vous assurer de pouvoir récupérer d'une telle erreur, vous pouvez utiliser la commande *COPY TABLE* (pour copier la définition de la table à supprimer vers un fichier de sauvegarde) avant d'exécuter la commande *DELETE*.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *DELETE* à partir de la liste déroulante.
2. Dans le champ de texte à droite, saisissez le nom de la table à supprimer.
3. Cliquez sur la flèche à droite pour ajouter la commande *DELETE TABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

EXPORT

La commande *EXPORT TABLE* est utilisée pour copier le contenu du Dictionnaire du Générateur vers un Fichier texte, nommé *xxxxxxx.tbl*, dans le répertoire DCT. Le nom de la table doit avoir une longueur de 6 caractères. La valeur "ALL" n'est pas permise.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *EXPORT* à partir de liste déroulante.
2. Dans le champ de texte à droite de la liste déroulante, saisissez le nom de la table dans laquelle vous voulez exporter le Dictionnaire du Générateur.

3. Cliquez sur la flèche à droite pour ajouter la commande *EXPORT TABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

IMPORT

La commande *IMPORT TABLE* est utilisée pour copier le contenu d'un Fichier texte, nommé *xxxxxxx.tbl*, vers le Dictionnaire du Générateur à partir du répertoire DCT. Le nom de la table doit avoir une longueur de 6 caractères. La valeur "ALL" n'est pas permise.

1. Sur la ligne *TABLE* de l'Assistant Commandes, sélectionnez *IMPORT* à partir de liste déroulante.
2. Dans le champ de texte à droite de la liste déroulante, saisissez le nom de la table à importer.
3. Cliquez sur la flèche à droite pour ajouter la commande *IMPORT TABLE* dans la fenêtre des commandes.
4. À ce stade, vous pouvez générer la commande composée ou vous pouvez ajouter des commandes additionnelles avant d'effectuer la génération.

Générer les fichiers MDL

MDL signifie MetaSuite Definition Language (Langage de définition MetaSuite). Ce langage de texte est utilisé pour définir les métadonnées (la structure des Enregistrements et des Champs) des Sources de données et des Cibles. Les Fichiers MDL qui contiennent les définitions des Enregistrements et des Champs d'une ou de plusieurs Sources de données ou Cibles sont appelés des "Fichiers de Dictionnaire".

Note: Si vous êtes familier avec le langage MDL, vous pouvez modifier manuellement les définitions MDL. Se référer au chapitre *Commandes du langage de définition* dans le Guide de l'Utilisateur MetaStore Manager. Cependant, cet usage n'est plus recommandé étant donné que vous pouvez faire toutes les modifications aux Fichiers de Dictionnaire en utilisant l'Interface Utilisateur MetaStore Manager.

Pour une description détaillée, se référer aux procédures suivantes:

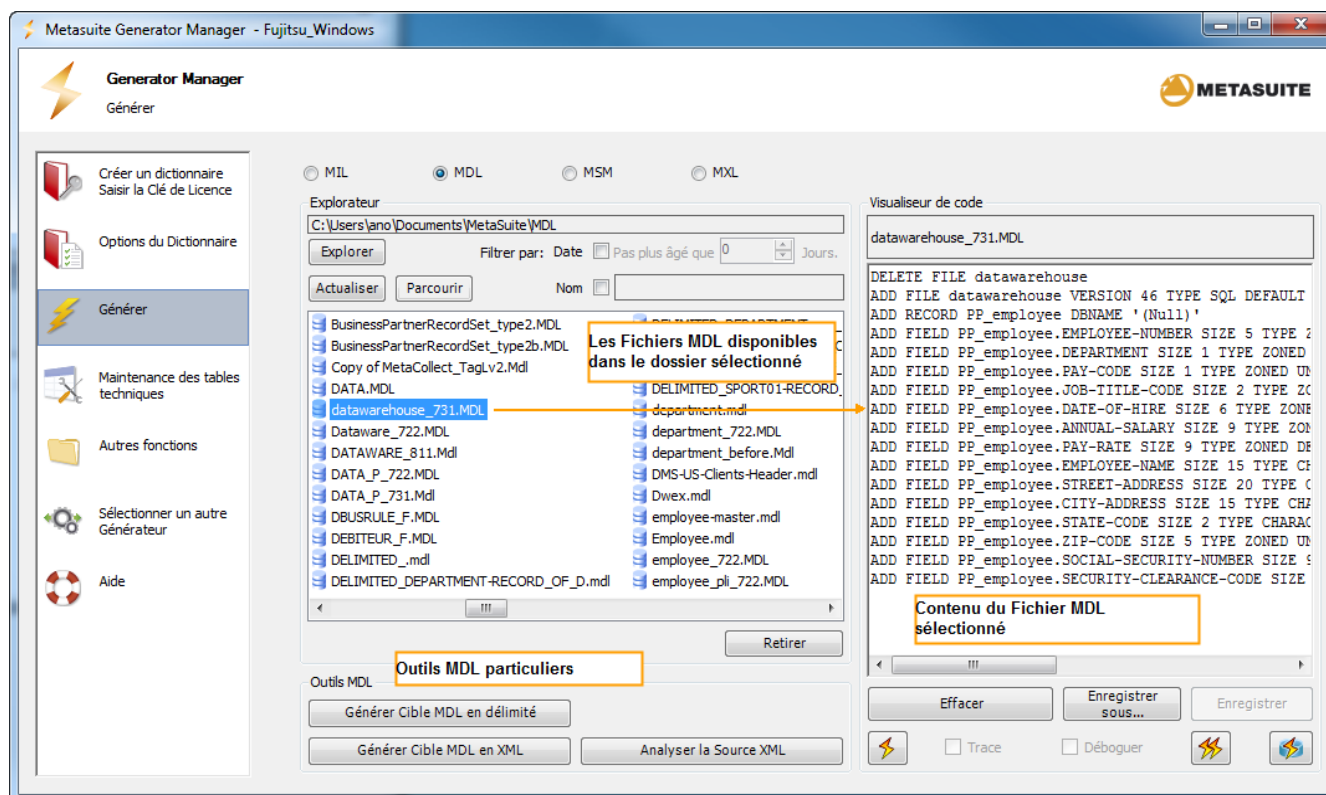
- [L'écran "Générer - MDL"](#) (page 38)
- [Éditer les Fichiers MDL existants](#) (page 39)
- [Utiliser les Outils MDL - Générer Cible MDL en délimité](#) (page 40)
- [Utiliser les Outils MDL - Générer Cible MDL en XML](#) (page 42)
- [Utiliser les Outils MDL - Analyser la Source XML](#) (page 45)

7.1. L'écran "Générer - MDL"

1. Cliquez sur le bouton *Générer* dans la colonne de gauche.

2. Sélectionnez l'option de format *MDL*.

L'écran suivant s'affiche:



Cet écran comprend trois sections:

Section	Description
La liste des Fichiers MDL disponibles	La liste s'affiche en haut à gauche. Sélectionnez un Fichier pour en afficher le contenu dans la fenêtre de contenu à droite. Voir Éditer les Fichiers MDL existants on page 39.
Outils MDL	Utilisez les Outils MDL pour les transformations MDL spécifiques: • Générer Cible MDL en délimité (page 40) • Générer Cible MDL en XML (page 42) • Analyser la Source XML (page 45)
Le panneau de contenu du Fichier MDL	Ce panneau contient la codification MDL contenu dans le fichier MDL sélectionné ou résultant de l'opération de l'outil MDL sélectionné. Vous pouvez manuellement éditer le code MDL affiché.

7.2. Éditer les Fichiers MDL existants

1. Accédez à l'écran *Générer MDL*.
Voir [L'écran "Générer - MDL"](#) on page 38.
2. Sélectionnez le Fichier MDL requis à partir de la liste des Fichiers MDL disponibles.
Son contenu s'affiche immédiatement dans le panneau de droite.

Note: Si vous sélectionnez un autre Fichier MDL, le contenu de ce fichier remplacera les instructions actuellement affichées. Vous pouvez supprimer un Fichier MDL en sélectionnant le fichier à partir de la liste des Fichiers MDL disponibles et en cliquant ensuite sur le bouton *Retirer*.

3. Si nécessaire, modifiez la codification MDL.
4. Une fois les modifications requises sont faites, vous pouvez exécuter une des actions suivantes:

Action	Description
Effacer	Efface le contenu de la liste pour pouvoir redémarrer.
Enregistrer sous	Sauvegarde le code MDL affiché dans un nouveau fichier MDL. Saisissez le nouveau Chemin et le nouveau Nom de fichier et cliquez sur le bouton OK.
Enregistrer	Remplace le Fichier MDL existant par le code MDL (modifié) affiché.



Cliquez sur le bouton *Générer* si vous voulez générer la codification MDL. La codification MDL sera générée pour que les nouveaux paramètres deviennent valides dans le Fichier MDL.

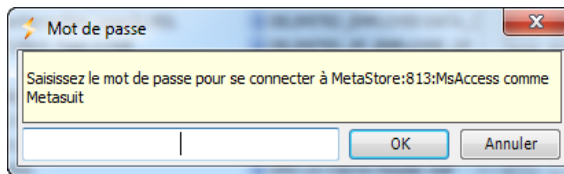
La fenêtre suivante confirme cette action:

```

Messages Generator Manager

MDL Génération de fichier
MDL Fichier généré avec avertissement(s)..

DELETE FILE datawarehouse
\FLS012W 01 datawarehouse Not defined in Dictionary
ADD FILE datawarehouse VERSION 46 TYPE SQL DEFAULT
ADD RECORD PP_employee DBNAME '(Null)'
ADD FIELD PP_employee.EMPLOYEE-NUMBER SIZE 5 TYPE ZONED UNSIGNED POSITION 1 RULE Employee identification
in the company
ADD FIELD PP_employee.DEPARTMENT SIZE 1 TYPE ZONED UNSIGNED POSITION 2 RULE Working department
ADD FIELD PP_employee.PAY-CODE SIZE 1 TYPE ZONED UNSIGNED POSITION 3 RULE paycode 1 weekly 2 monthly
ADD FIELD PP_employee.JOB-TITLE-CODE SIZE 2 TYPE ZONED UNSIGNED POSITION 4
ADD FIELD PP_employee.DATE-OF-HIRE SIZE 6 TYPE ZONED UNSIGNED POSITION 5 RULE started working at
ADD FIELD PP_employee.ANNUAL-SALARY SIZE 9 TYPE ZONED DECIMAL 2 UNSIGNED POSITION 6
ADD FIELD PP_employee.PAY-RATE SIZE 9 TYPE ZONED DECIMAL 2 UNSIGNED POSITION 7
ADD FIELD PP_employee.EMPLOYEE-NAME SIZE 15 TYPE CHARACTER POSITION 8
ADD FIELD PP_employee.STREET-ADDRESS SIZE 20 TYPE CHARACTER POSITION 9
ADD FIELD PP_employee.CITY-ADDRESS SIZE 15 TYPE CHARACTER POSITION 10
ADD FIELD PP_employee.STATE-CODE SIZE 2 TYPE CHARACTER POSITION 11
ADD FIELD PP_employee.ZIP-CODE SIZE 5 TYPE ZONED UNSIGNED POSITION 12
ADD FIELD PP_employee.SOCIAL-SECURITY-NUMBER SIZE 9 TYPE ZONED UNSIGNED POSITION 13
ADD FIELD PP_employee.SECURITY-CLEARANCE-CODE SIZE 5 TYPE ZONED UNSIGNED POSITION 14
\
WARNING LEVEL ERRORS
1
STARTED AT 13/04/17 10:54:14 - STOPPED AT 13/04/17 10:54:14 - ELAPSED TIME 0.01 SEC.
  
```

Action	Description
	Générer plusieurs Fichiers MDL à la fois. Sélectionnez les fichiers que vous voulez générer et cliquez sur le bouton <i>Générer</i> .
	Importe les Fichiers de codification MDL en mode batch. La fenêtre suivante s'affiche:  Saisissez le mot de passe et cliquez sur le bouton OK. Le Journal sera affiché.

7.3. Utiliser les Outils MDL - Générer Cible MDL en délimité

Les Fichiers MDL Cible délimités ou séparés par des virgules sont des Fichiers séquentiels qui contiennent des valeurs de champ, chaque valeur étant séparée par un point-virgule ou un autre caractère. Certains fichiers délimités contiennent également un Caractère de Code fin de rangée.

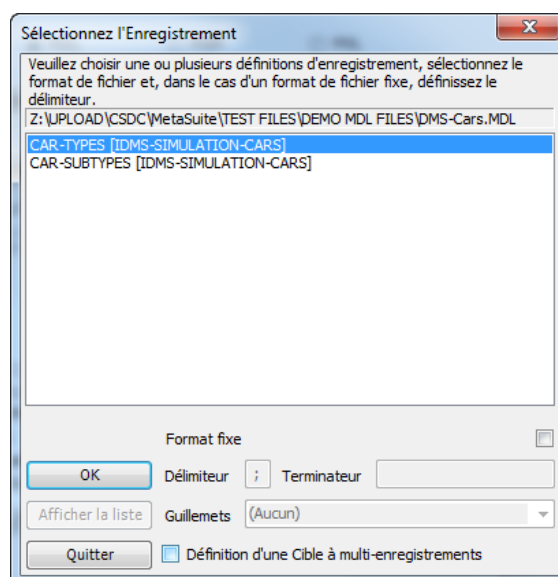
En raison de leur portabilité, ils peuvent être utiles si vous voulez faire des conversions vers un autre système d'exploitation.

1. Accédez à l'écran MDL.

Voir [L'écran "Générer - MDL"](#) on page 38.

2. À partir de la liste des fichiers disponibles, sélectionnez le Fichier MDL pour lequel vous voulez créer une Cible MDL en délimité.
3. Sélectionnez l'option *Générer Cible MDL en délimité* à partir de la section "Outils MDL".

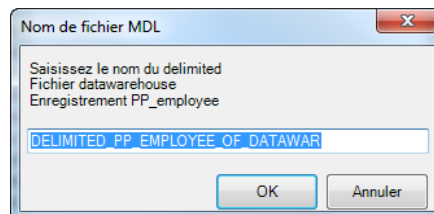
La fenêtre suivante s'affiche:



4. En haut de la fenêtre, sélectionnez un ou plusieurs Enregistrements qui doivent apparaître dans la Cible MDL en délimité.
5. Spécifier les options requises dans la section en-dessous de la liste des Enregistrements.
Les options suivantes sont disponibles:

Option	Description
Format fixe	Cochez cette case si vous voulez un format fixe. Chaque champ aura un point de départ prédéfini. Décochez cette case si vous avez besoin d'un format de sortie variable. Les champs seront placés l'un derrière l'autre sans espaces entre eux.
Délimiteur	Dans ce champ vous pouvez écraser le Délimiteur par défaut, si nécessaire. Le caractère par défaut est un point-virgule.
Termineur	Dans ce champ vous pouvez écraser le Termineur par défaut, si nécessaire. La valeur par défaut est une chaîne de caractères vide.
Guillemets	Dans ce champ vous pouvez écraser les Guillemets par défaut, si nécessaire. Vous pouvez entourer les chaînes de caractères alphanumériques de guillemets. Ceci est utile si les espaces et les caractères du Délimiteur ou Code fin de rangée peuvent faire partie de la chaîne de caractères et ne devraient pas être reconnus comme tel.
Définition d'une Cible à multi-enregistrements	Cochez cette case si vous voulez créer une seule définition de Fichier MDL contenant toutes les définitions d'Enregistrements. Le Fichier Cible sera également un Fichier à multi-enregistrements. Décochez cette case si vous voulez créer une définition de Fichier par définition d'Enregistrement. Chaque Fichier Cible contiendra une seule définition d'Enregistrement.

6. Une fois que vous avez fait toutes les sélections nécessaires, cliquez sur le bouton OK.
La fenêtre suivante s'affiche:



Le nom suggéré par défaut est composé comme suit:

DELIMITED[_RECORDNAME]_FILENAME

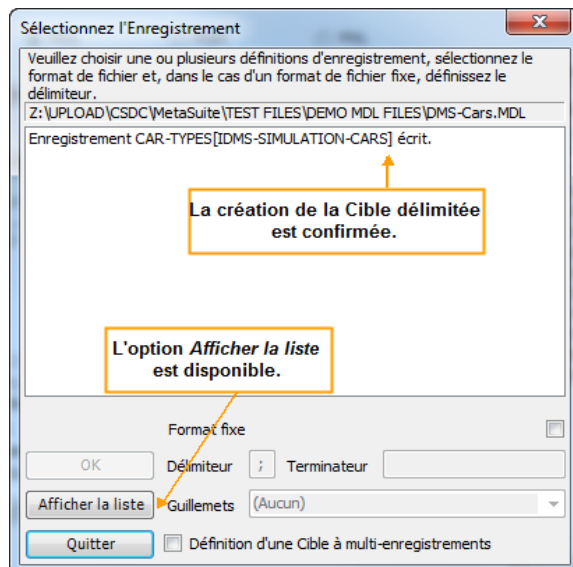
Note:

- Le "RECORDNAME" n'est pas disponibles pour les Fichiers Cible multi-enregistrements.
- La longueur maximale du Fichier est 32 caractères. Les noms plus longs seront tronqués.
- L'extension MDL est ajoutée automatiquement.
- La Cible MDL en délimité générée est sauvegardée dans le dossier MDL par défaut, comme l'illustre le Chemin en haut de l'écran.

7. Si nécessaire, modifier le nom par défaut, et cliquez sur le bouton OK.

Résultats:

- La Cible MDL en délimité sera générée et elle sera affichée dans la liste des Fichiers MDL disponibles.
- La fenêtre *Sélectionnez l'Enregistrement* est affichée à nouveau:



8. Cliquez sur le bouton *Afficher la liste* pour afficher un rapport dans le format suivant:

datawarehouse_731

[Enregistrement précédent](#)

[Enregistrement suivant](#)

N°	Name	Size	Type	Description
1	EMPLOYEE-NUMBER	5	numeric	Employee identification in the company
2	DEPARTMENT	1	numeric	Working department
3	PAY-CODE	1	numeric	paycode 1 weekly 2 monthly
4	JOB-TITLE-CODE	2	numeric	-
5	DATE-OF-HIRE	6	numeric	started working at
6	ANNUAL-SALARY	9	numeric	-

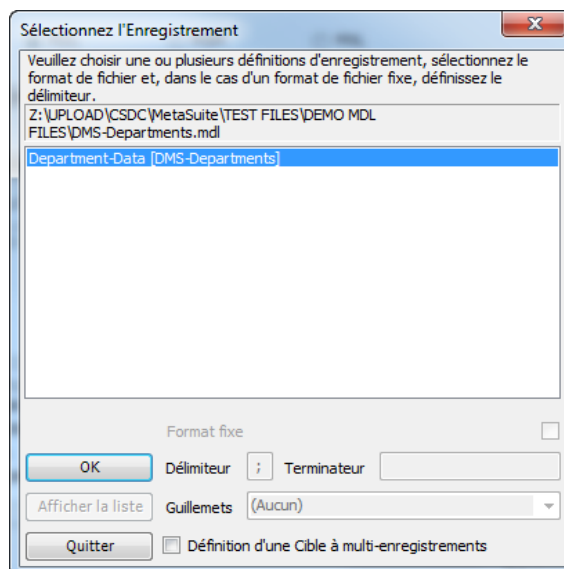
7.4. Utiliser les Outils MDL - Générer Cible MDL en XML

Les Cibles MDL et XML sont des Fichiers Cible séquentiels dont le contenu est en format XML.

1. Accédez à l'écran *MDL*.
Voir [L'écran "Générer - MDL"](#) on page 38.
2. À partir de la liste des fichiers disponibles, sélectionnez le Fichier MDL pour lequel vous voulez créer la Cible MDL en XML.

3. Sélectionnez l'option *Générer Cible MDL en XML* à partir de la section "Outils MDL".

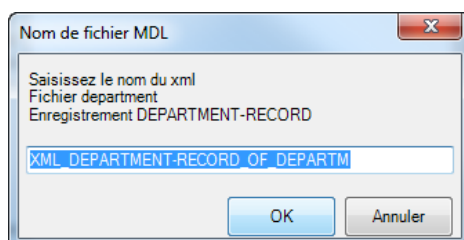
L'écran suivant s'affiche:



4. En haut de la fenêtre, sélectionnez un ou plusieurs Enregistrements qui doivent apparaître dans la Cible MDL en XML.
5. Spécifier les options requises dans la section en-dessous de la liste des Enregistrements. Les options suivantes sont disponibles:

Option	Description
Format fixe	Cette option n'est pas disponible pour les Cibles MDL en XML.
Délimiteur	Cette option n'est pas disponible pour les Cibles MDL en XML.
Termineur	Cette option n'est pas disponible pour les Cibles MDL en XML.
Guillemets	Cette option n'est pas disponible pour les Cibles MDL en XML.
Définition d'une Cible à multi-enregistrements	Cochez cette case si vous voulez créer une seule définition de Fichier MDL contenant toutes les définitions d'Enregistrements. Décochez cette case si vous voulez créer une définition de Fichier MDL séparée pour chaque définition d'Enregistrement.

6. Une fois que vous avez fait toutes les sélections nécessaires, cliquez sur le bouton OK. La fenêtre suivante s'affiche:



Le nom suggéré par défaut est composé comme suit:

XML[_RECORDNAME_OF_]FILENAME

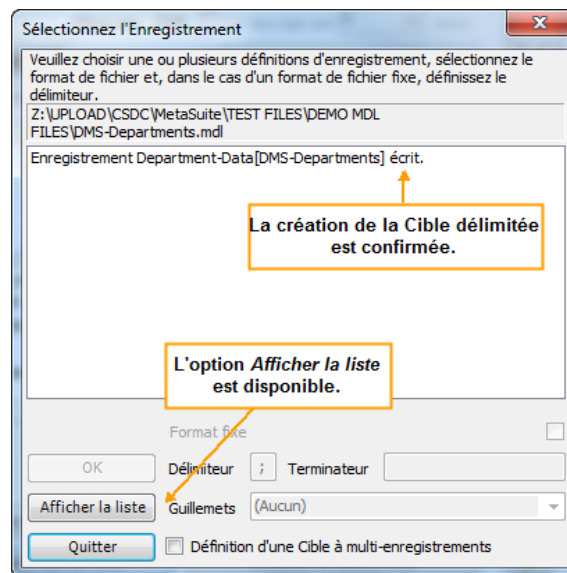
Note:

- Le "RECORDNAME_OF" n'est pas disponibles pour les Fichiers Cible multi-enregistrements.
- La longueur maximale du Fichier est 32 caractères. Les noms plus longs seront tronqués.
- L'extension MDL est ajoutée automatiquement.
- La Cible MDL en délimité générée est sauvegardée dans le dossier MDL par défaut, comme l'illustre le Chemin en haut de l'écran.

7. Si nécessaire, modifiez le nom par défaut, et cliquez sur le bouton OK.

Résultats:

- La Cible MDL en XML sera générée et elle sera affichée dans la liste des Fichiers MDL disponibles.
- La fenêtre *Sélectionnez l'Enregistrement* est affichée à nouveau:



8. Cliquez sur le bouton *Afficher la liste* pour afficher un rapport dans le format suivant:

department

[Enregistrement précédent](#)

[Enregistrement suivant](#)

N°	Name	Size	Type	Description
1	FIELD_1	103	character	-

7.5. Utiliser les Outils MDL - Analyser la Source XML

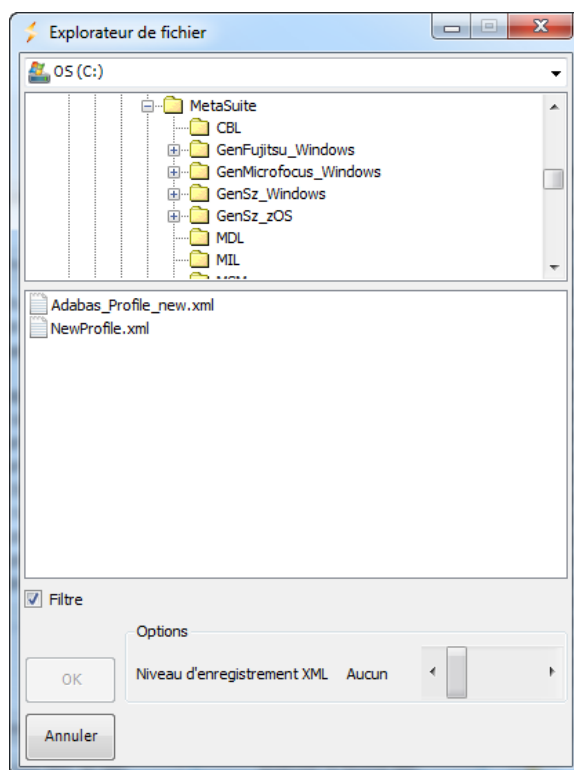
Les Fichiers Source XML peuvent être convertis en MDL pour qu'ils puissent être utilisés dans MetaSuite.

1. Accédez à l'écran MDL.

Voir [L'écran "Générer - MDL"](#) on page 38.

2. Sélectionnez l'option *Analyser la Source XML* à partir de la section "Outils MDL".

L'écran suivant s'affiche:



3. Accédez le dossier requis et sélectionnez le Fichier requis à partir de la liste.

Note: Cochez la case *Filtre* pour n'afficher que les Fichiers XML.

4. Spécifier les options requises dans la section en-dessous de la liste des Fichiers.

Les options suivantes sont disponibles:

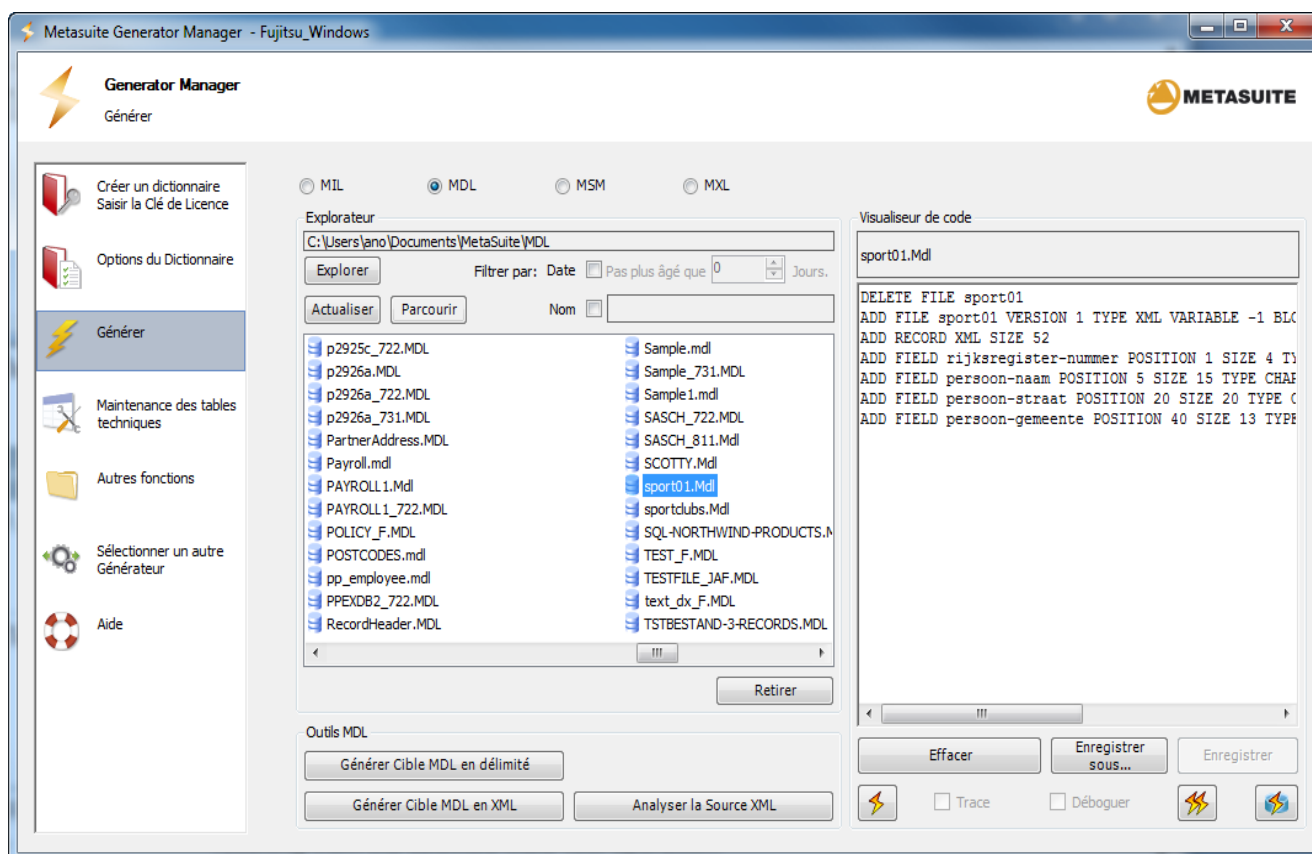
Option	Description
Analyser les [nombre] premiers Enregistrements	Cochez cette case pour n'analyser qu'un nombre spécifié d'Enregistrements au début du Fichier. Si vous cochez cette case, vous devez également fournir le nombre d'Enregistrements à analyser.
Délimiteur	Dans ce champ vous pouvez écraser le Délimiteur par défaut, si nécessaire.

Option	Description
Code fin de rangée	Dans ce champ vous pouvez écraser le Code fin de rangée par défaut si nécessaire.
Niveau d'enregistrement XML	Sélectionnez le niveau requis: - Sélectionnez 0 si aucun niveau XML ne doit être pris en considération. Les Champs XML seront analysés, quel que soit le niveau de leur balise XML. Cela implique que tous les Champs doivent avoir un nom unique. - Sélectionnez 1 si tous les niveaux XML doivent être pris en considération. Le premier niveau XML sera sélectionné comme "niveau de l'Enregistrement". Les balises à ce niveau contiendront le nom de l'Enregistrement. - Sélectionnez 2 si les balises XML de niveau 2 ou supérieur doivent être prises en considération. Les balises au premier niveau seront ignorées. Les balises XML de niveau 2 seront sélectionnées comme "niveau de l'Enregistrement". - Si vous sélectionnez une valeur plus élevée, un niveau plus élevé sera considéré comme "niveau de l'Enregistrement".

5. Une fois que vous avez fait toutes les sélections nécessaires, cliquez sur le bouton OK.

Le Fichier Source XML est analysé et le Fichier MDL résultant est ajouté à la liste des Fichiers MDL disponibles. Le nom du Fichier est sauvegardé, seule l'extension est modifiée en MDL.

Exemple de fichier XML:



6. Si nécessaire, éditer le Fichier MDL.

Générer les fichiers MSM

L'abréviation MSM est utilisée pour indiquer les Fichiers de Modèle MetaSuite (MetaSuite Model). Ces fichiers contiennent les définitions des Sources de données, des Cibles de données et les règles de mappage entre eux dans un *format codifié*.

Vous pouvez éditer les Modèles MetaSuite en utilisant l'application MetaMap Manager.

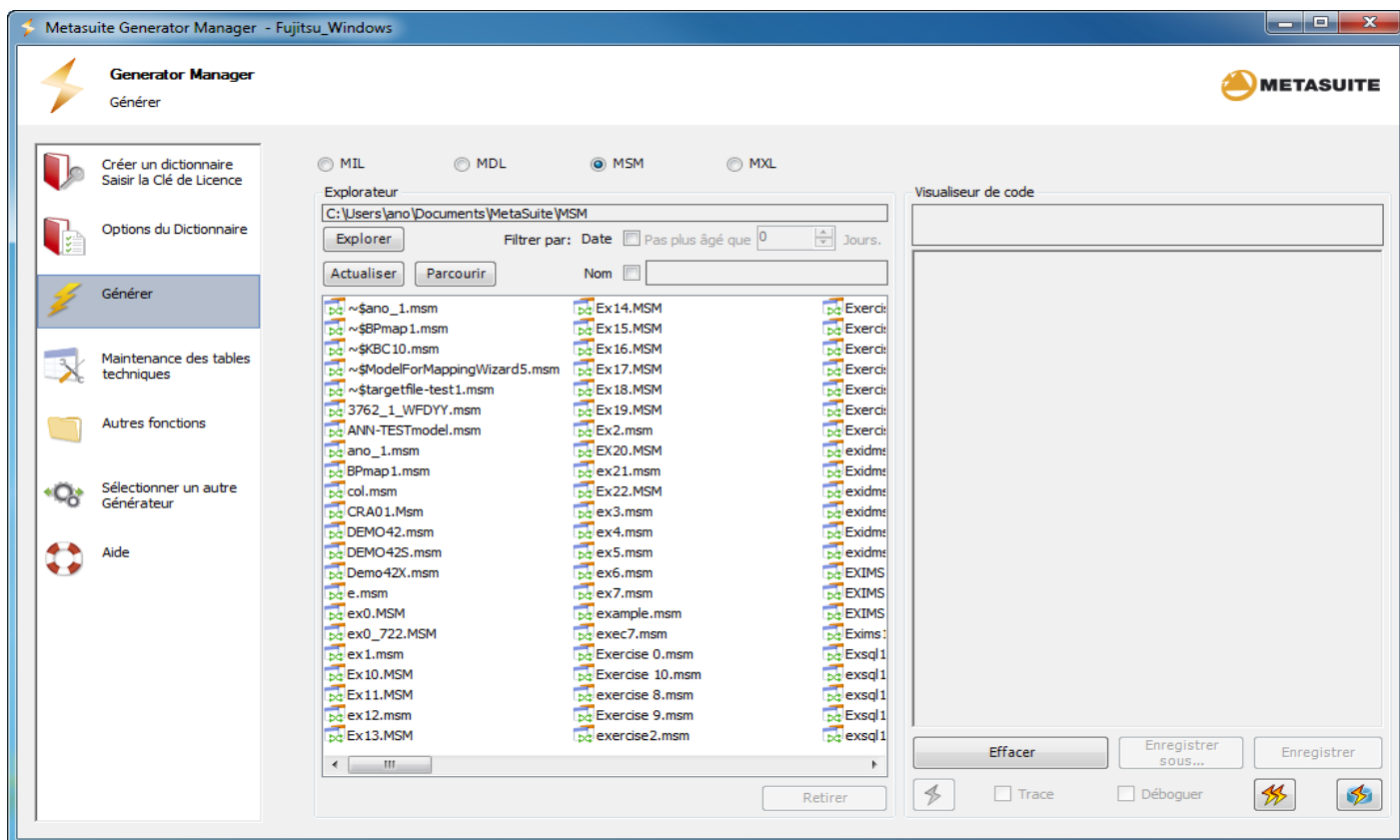
Le Modèle MetaSuite existe également en version texte, appelée Langage d'eXtraction MetaSuite (MXL). Voir [Générer les fichiers MXL](#) on page 49.

Les Fichiers MSM peuvent être exportés en mode batch. Suivez la procédure décrite ci-dessous:

1. Cliquez sur le bouton *Générer* dans la colonne de droite.

2. Sélectionnez l'option de format *MSM*.

L'écran suivant s'affiche:



Les Modèles disponibles sont affichés à gauche.

3. Deux options sont disponibles:

3.1. Génération multiple en Batch

Si vous devez générer plusieurs Fichiers MSM à la fois, cliquez sur le bouton *Génération multiple en Batch* ()

Sélectionnez les Fichiers que voulez générer et cliquez sur le bouton *Générer*.

3.2. Exporter le Modèle MetaSuite en mode Batch

Pour exporter le Modèle MetaSuite en mode batch, cliquez sur le bouton *Exporter le Modèle*

MetaSuite en mode Batch ()

Le Fichier MXL correspondant sera généré.

Générer les fichiers MXL

L'abréviation MXL est utilisée pour indiquer les Fichiers de Langage d'eXtraction MetaSuite (MetaSuite eXtraction Language). Ces fichiers contiennent les définitions des Sources de données, des Cibles de données et les règles de mappage entre eux dans un *format texte*.

Les Fichiers MXL peuvent être générés. Voir [Générer un Fichier MXL](#) on page 50.

Il est également possible de générer plusieurs Fichiers MXL en Mode batch. Ceci est intéressant dans le cas où une nouvelle construction du Générateur est installée et que vous voulez régénérer certains ou tous les fichiers MXL. Voir [Générer plusieurs Fichiers MXL en mode batch](#) on page 52.

Pour une description détaillée, se référer aux procédures suivantes:

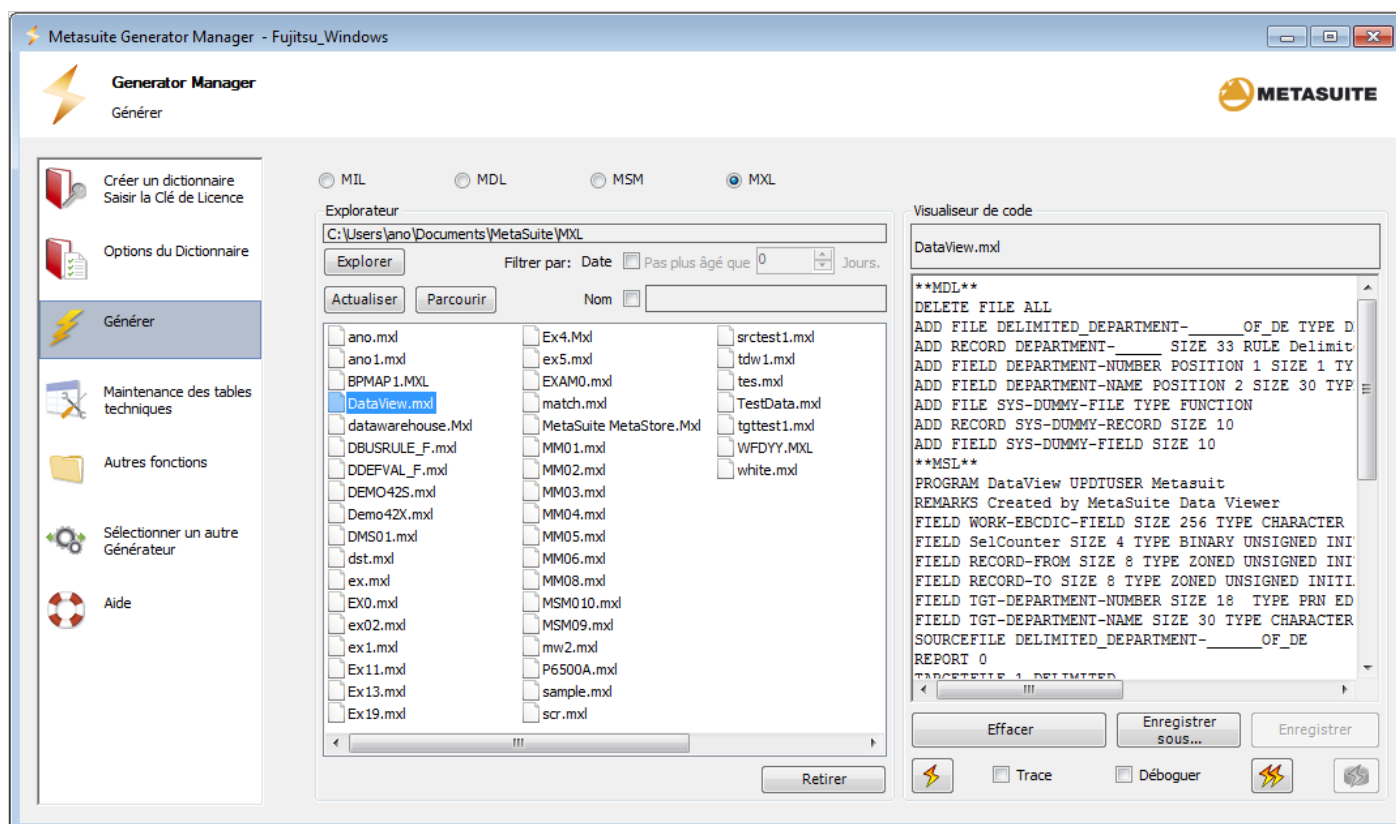
- [Générer un Fichier MXL](#) (page 50)
- [Générer plusieurs Fichiers MXL en mode batch](#) (page 52)

9.1. Générer un Fichier MXL

1. Cliquez sur le bouton *Générer* dans la colonne de gauche.

2. Sélectionnez l'option de format *MXL*.

L'écran suivant s'affiche:



Les Fichiers MXL sont affichés à gauche.

3. Sélectionnez le Fichier MXL requis.

Son contenu s'affichera dans la fenêtre à droite.

Note: Si vous êtes familier avec le langage MXL, vous pouvez modifier directement le code. Pour des informations plus détaillées, se référer au chapitre *Commandes du langage de définition* dans le *Guide de l'Utilisateur de MetaStore Manager*. Cependant, nous vous conseillons de faire les modifications dans MetaMap Manager.

4. Une fois les modifications requises faites, vous pouvez exécuter une des actions suivantes:

Action	Description
Effacer	Retire toute la codification de sorte que vous pouvez recommencer de zéro.
Enregistrer sous...	Enregistre la codification dans son format courant dans un nouveau Fichier MXL.
Enregistrer	Enregistre la codification dans son format courant dans le Fichier MXL existant.

Action

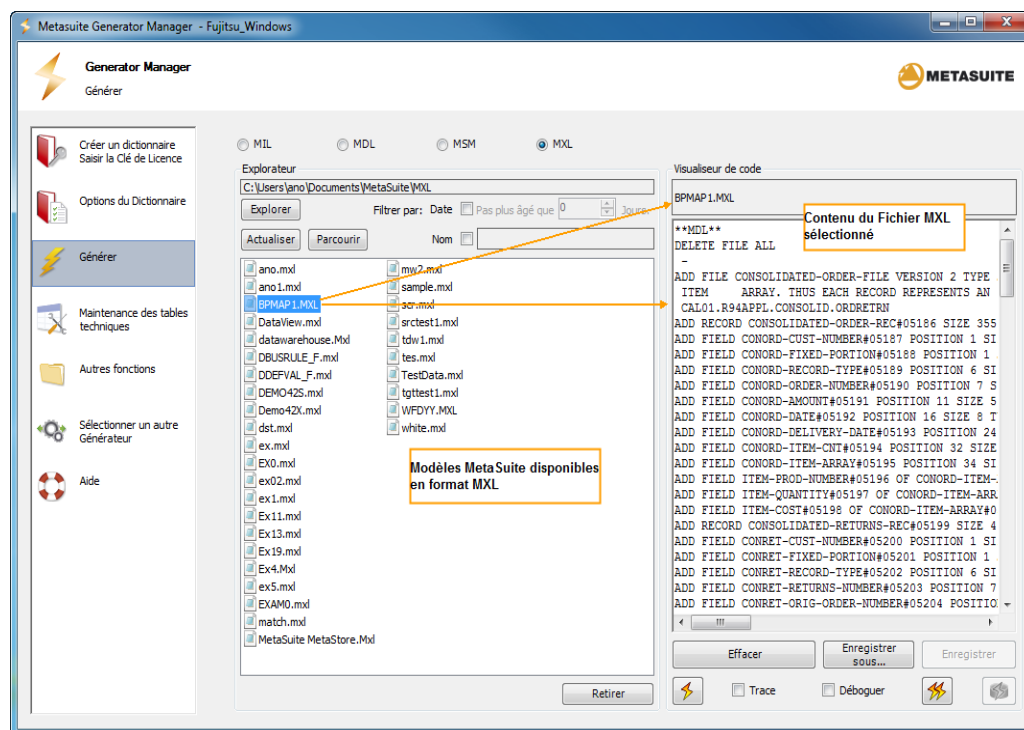
Description



Génère le Fichier MXL. Cela s'avère intéressant pour vérifier si le fichier contient des erreurs de syntaxe.

Vous pouvez cocher les cases "TRACE" et/ou "DEBUG (DÉBOGUER)" pour inclure des informations additionnelles.

Sauf si vous avez spécifié d'autres dossiers par défaut dans INI Manager, le fichier MGL résultant (programme COBOL) et le fichier MRL seront enregistrés dans les dossiers MGL et MRL du répertoire du Generator Manager utilisé.



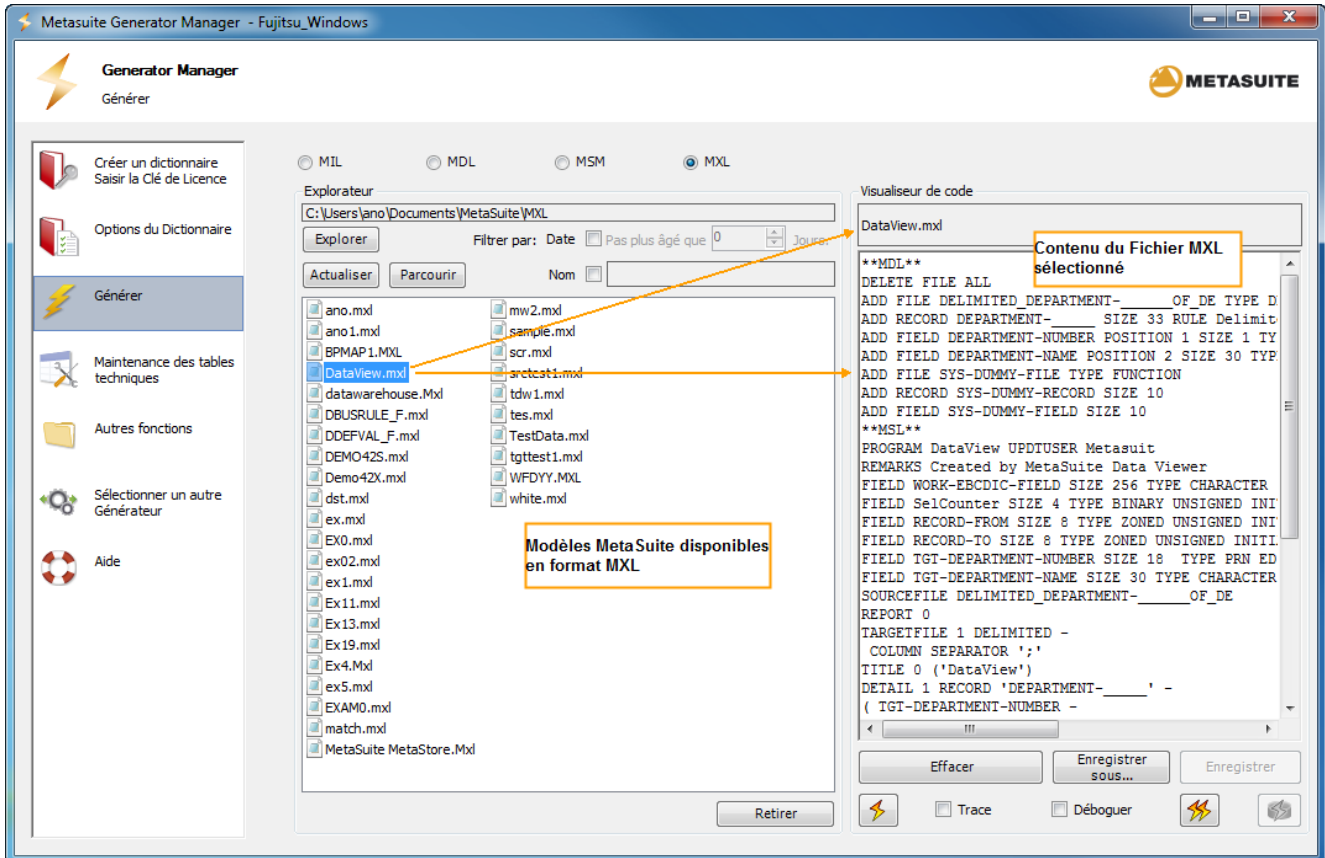
Génère plusieurs Fichiers MXL à la fois.

Sélectionnez les fichiers que vous voulez générer et cliquez sur le bouton **Générer**.

9.2. Générer plusieurs Fichiers MXL en mode batch.

1. Cliquez sur le bouton *Générer* dans la colonne de gauche.
2. Sélectionnez l'option de format *MXL*.

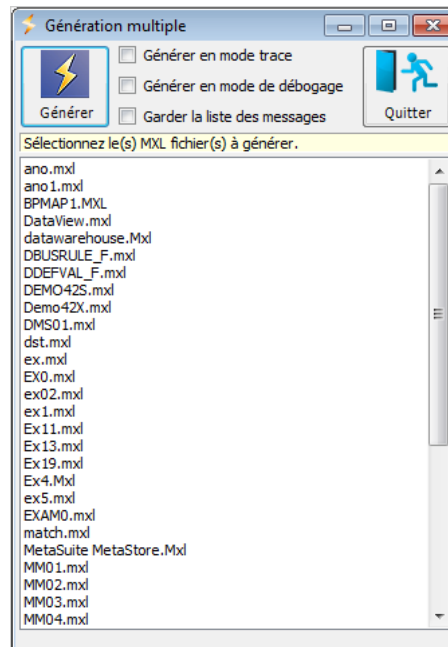
L'écran suivant s'affiche:



Les Modèles disponibles sont affichés à gauche.

3. Cliquez sur le bouton .

La fenêtre suivante s'affiche:



4. Sélectionnez les Fichiers que vous voulez générer en mode batch et cliquez sur le bouton *Générer*.

Vous pouvez cocher les cases "TRACE" et/ou "DEBUG (DÉBOGUER)" pour inclure des informations additionnelles.

Sauf si vous avez spécifié d'autres dossiers par défaut dans INI Manager, le fichier MGL résultant (programme COBOL) et le fichier MRL seront enregistrés dans les dossiers MGL et MRL dans le répertoire racine du Generator Manager utilisé.

Maintenance des tables techniques

- Aperçu

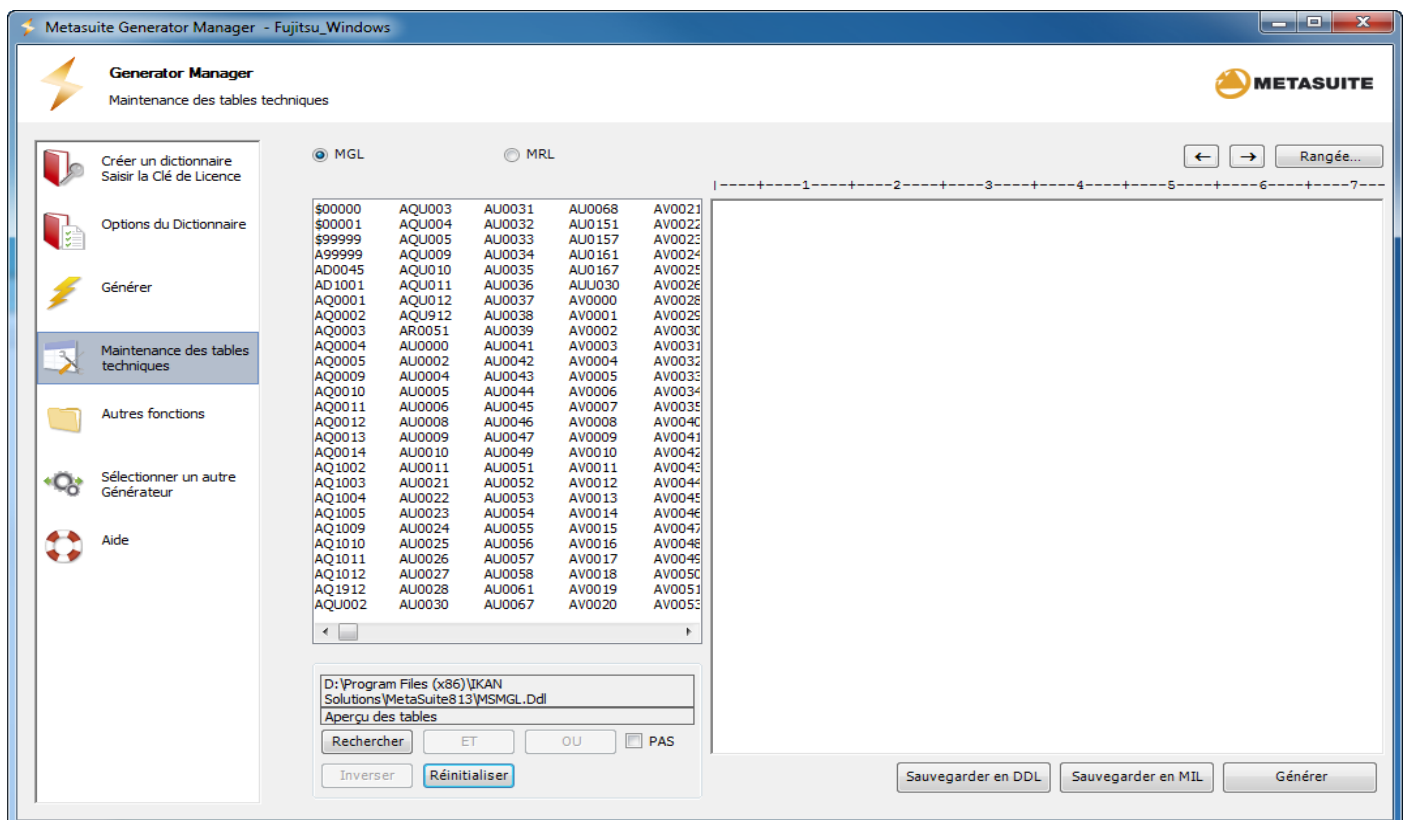
L'écran *Maintenance des tables techniques* MGL et MRL.

Pour plus d'informations, se référer aux chapitres spécifiques:

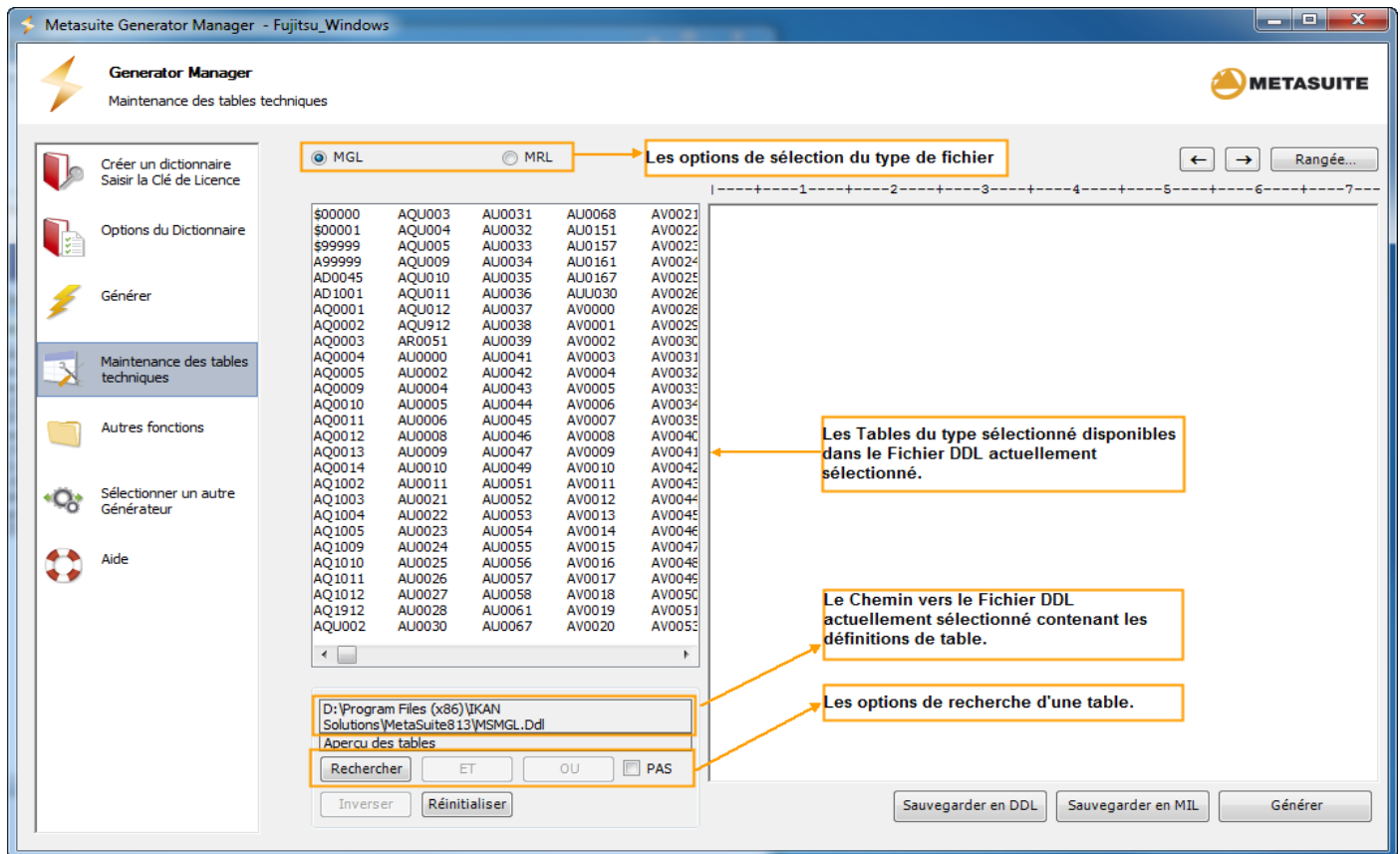
- [Maintenance des tables techniques - Maintenance des tables MGL](#) (page 58)
- [Maintenance des tables techniques - Maintenance des tables MRL](#) (page 78)

1. Accédez à la fenêtre *Maintenance des tables techniques* en cliquant sur le bouton *Maintenance des tables techniques* dans la colonne de gauche.

L'écran suivant s'affiche:



Les éléments en commun disponibles sont affichés à gauche:



Note: Les autres éléments sont spécifiques pour chaque type de Fichier Pour plus d'informations, se référer aux chapitres spécifiques.

2. Sélectionnez le type de fichier requis.

Les types suivants sont disponibles:

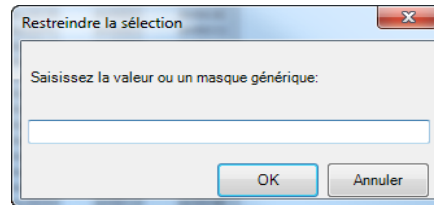
- [MGL](#) (page 58)
- [MRL](#) (page 78)

3. Vérifier le chemin et le nom de Fichier affichés en dessous de la liste des tables disponibles. Il réfère au Fichier DDL contenant la définition des tables affichées.

4. Si vous ne trouvez pas immédiatement la table requise dans la liste des tables disponibles, vous pouvez utiliser les options de recherche d'une table.

Ces options permettent de vérifier des chaînes de caractères sont disponibles dans le *contenu* des Tables. Cependant, si chaque table contient son nom sur la première ligne, vous pouvez les utiliser également pour trouver des noms de table spécifiques.

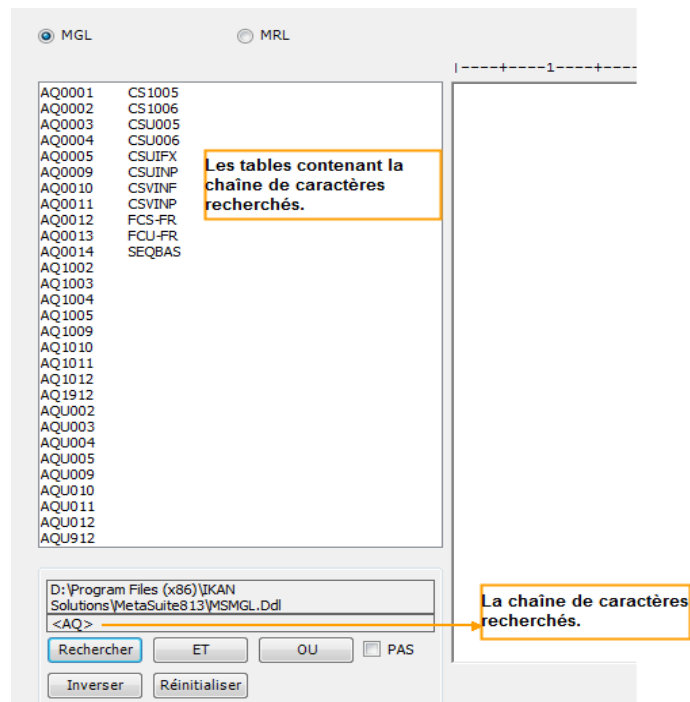
Option	Description
Rechercher	Cliquez sur le bouton <i>Rechercher</i> pour spécifier une chaîne de caractères recherchés. L'écran suivant s'affiche:



Définissez la chaîne de caractères recherchés et cliquez sur le bouton OK.

Résultat:

- La chaîne de caractères recherchés s'affiche au-dessus des options de recherche.
- La liste des tables affichées est limitée aux tables contenant la chaîne de caractères recherchés.



AND

Cette option permet de combiner deux chaînes de caractères recherchés. Les tables contenant LES DEUX chaînes de caractères recherchés seront affichées.

- Utilisez l'option *Rechercher* pour définir la chaîne de caractères recherchés.
- Cliquez sur le bouton 4 et saisissez la deuxième chaîne de caractères recherchés

Résultat:

- La chaîne de caractères recherchés s'affiche au-dessus des options de recherche.
- La liste des tables affichées est limitée aux tables contenant les deux chaînes de caractères recherchés.

Option	Description
OU	Cette option permet de combiner deux chaînes de caractères recherchés. Les tables contenant une ou les deux chaînes de caractères recherchés seront affichées. • Utilisez l'option <i>Rechercher</i> pour définir la chaîne de caractères recherchés. • Cliquez sur le bouton <i>OU</i> et saisissez la deuxième chaîne de caractères recherchés. Résultat: • La chaîne de caractères recherchés s'affiche au-dessus des options de recherche. • La liste des tables affichées est limitée aux tables contenant une ou les deux chaînes de caractères recherchés.
PAS	Cette option permet de nier une chaîne de caractères recherchés spécifiée. Les tables ne contenant PAS la chaîne de caractères recherchés seront affichées. • Cochez la case <i>PAS</i>. • Cliquez sur le bouton <i>Rechercher</i> et saisissez la chaîne de caractères recherchés à nier. • Cliquez sur le bouton <i>OK</i>. Résultat: • La chaîne de caractères recherchés s'affiche au-dessus des options de recherche. • La liste des tables affichées est limitée aux tables ne contenant PAS la chaîne de caractères recherchés spécifiée.
Inverser	Cette option permet d'inverser une chaîne de caractères recherchés combinée. Les tables qui ne correspondent PAS à la chaîne de caractères recherchés spécifiée seront affichées. • Définissez la chaîne de caractères recherchés, en utilisant les options "Rechercher", "ET" et "OU". • Cliquez sur le bouton <i>Inverser</i> pour inverser la chaîne de caractères recherchés complète. Résultat: • La chaîne de caractères recherchés s'affiche au-dessus des options de recherche. • La liste des tables affichées est limitée aux tables ne correspondant pas à la chaîne de caractères recherchés spécifiée.
Réinitialiser	Cliquez sur le bouton <i>Réinitialiser</i> pour effacer les critères de sélection spécifiés. Toutes les tables seront affichées à nouveau.

Maintenance des tables techniques

- Maintenance des tables MGL

L'abréviation MGL signifie MetaSuite Generate Language (Langage de génération MetaSuite). Le *Fichier MSMGL.DDL* (placé dans le dossier d'Installation MetaSuite) est un des trois Fichiers chargés pour le Dictionnaire du Générateur. Le Fichier *MSMGL.DDL* contient les tables MGL composées de petits morceaux de code COBOL, requis pour le bon fonctionnement du Générateur.

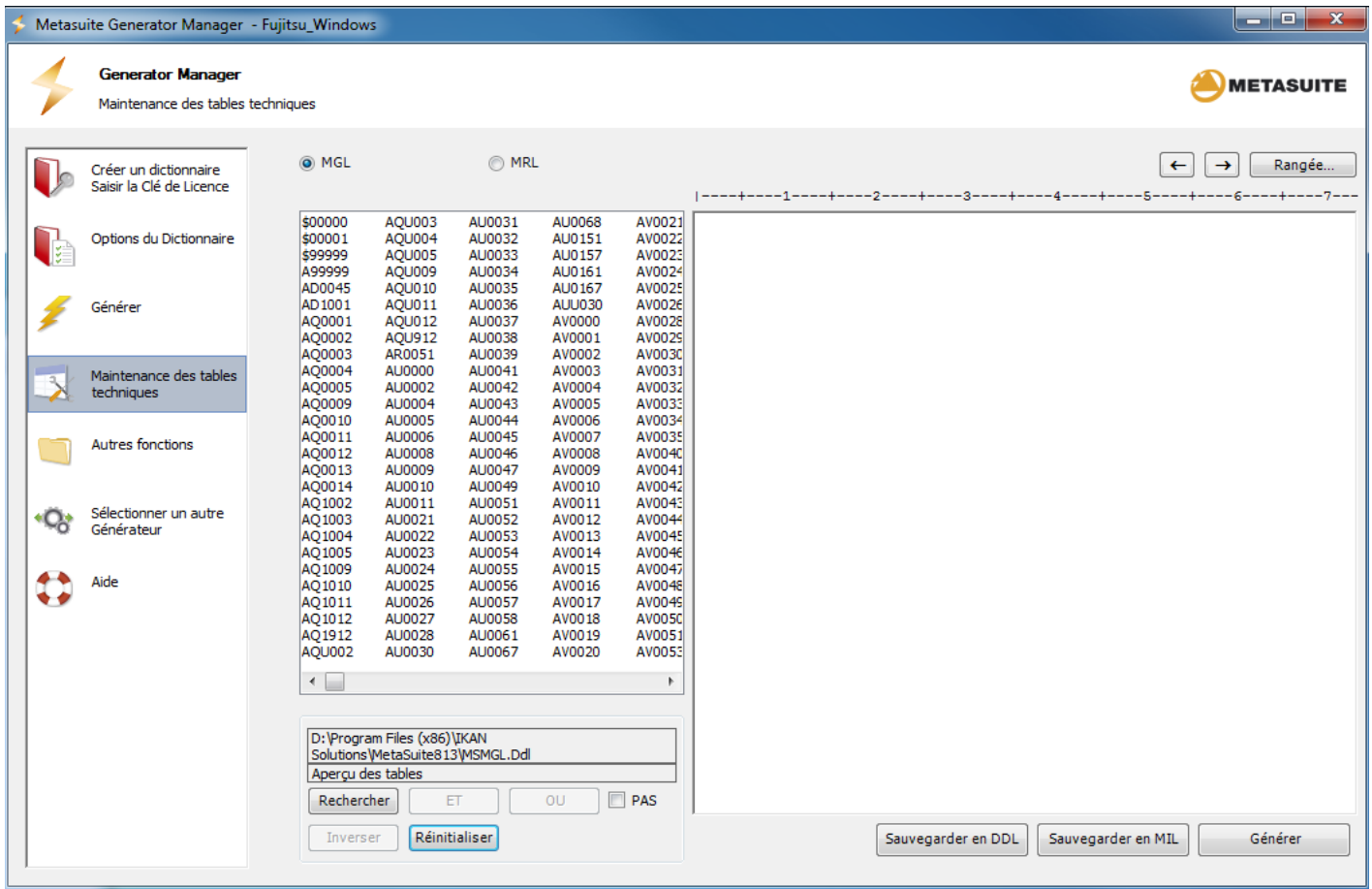
Il est logique de créer un nouveau Dictionnaire après avoir édité les tables MGL (uniquement pour les Utilisateurs expérimentés!).

Pour une description détaillée, se référer aux procédures suivantes:

- [L'écran de Maintenance des tables techniques MGL](#) (page 59)
- [Tables MGL: Description détaillée](#) (page 64)
- [Tables de Code-Contrôle](#) (page 65)
- [Tables de code COBOL](#) (page 74)
- [Exemples de table de code](#) (page 74)
- [Créer des Tables de Code-Contrôle personnalisées](#) (page 76)
- [Créer des Tables de code COBOL personnalisées](#) (page 76)

11.1. L'écran de Maintenance des tables techniques MGL

1. Cliquez sur le bouton *Maintenance des tables techniques* dans la colonne de gauche.
L'écran suivant s'affiche:

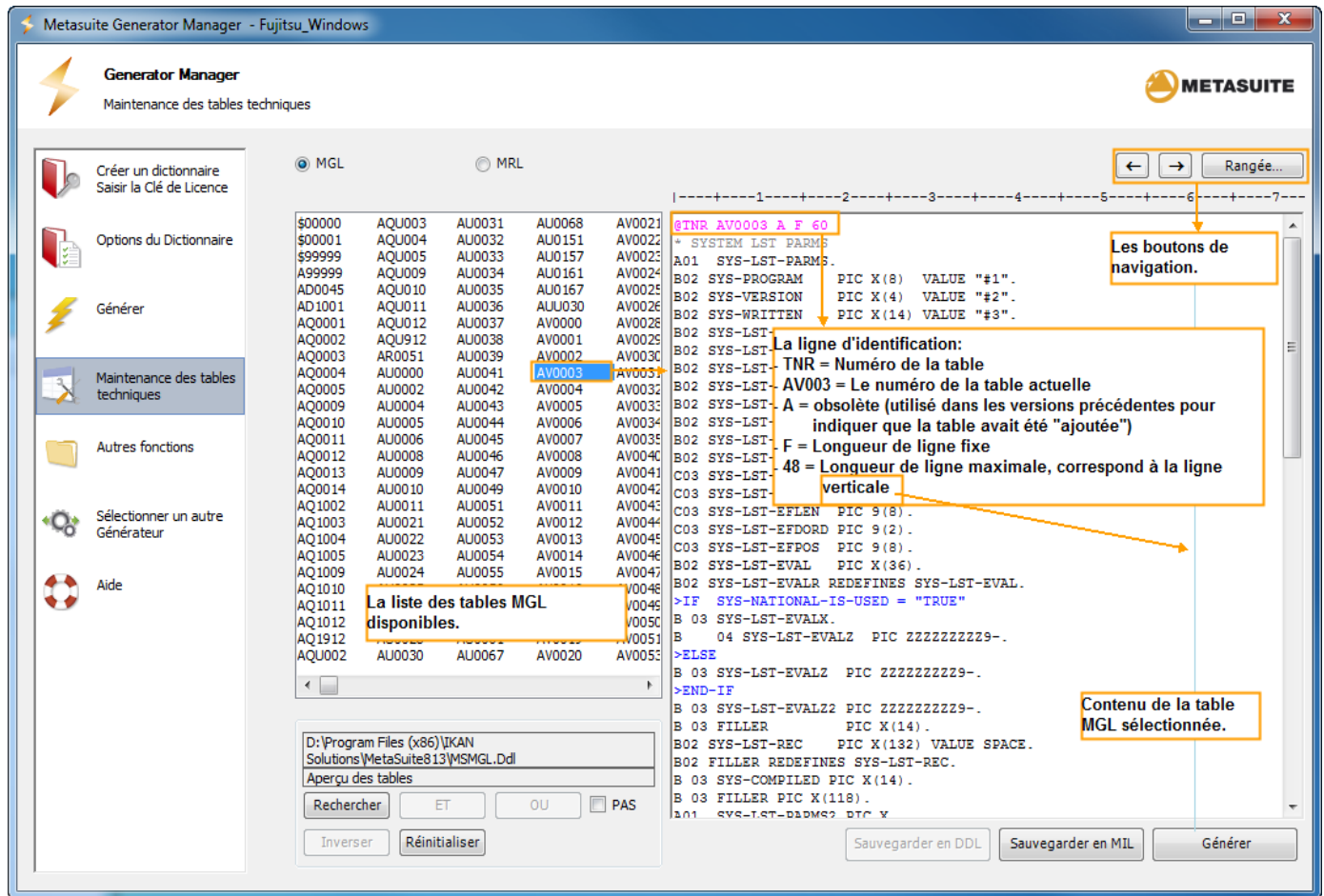


2. Sélectionnez l'option de format *MGL*.
Les tables MGL disponibles sont affichées.

Note: Si vous voulez limiter la liste des tables MGL affichées, vous pouvez définir des chaînes de caractères recherchés comme illustré dans le chapitre [Maintenance des tables techniques - Aperçu](#) (page 54).

3. Cliquez sur une table MGL pour en afficher son contenu à droite.

L'écran suivant s'affiche:

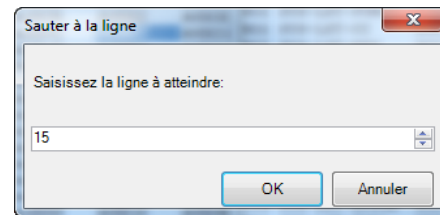


4. Si nécessaire, utilisez les boutons de navigation.

Les boutons suivants sont disponibles:

Icône	Option	Description
	Table précédente	Si vous avez affiché une séquence de tables, vous pouvez sélectionner cette option pour revenir en arrière dans la séquence et pour réafficher la table MGL précédente. En cliquant plusieurs fois sur cette option, vous pouvez revenir jusqu'à 10 tables en arrière. S'il n'y a plus de table précédente à afficher, cette option est désactivée.

Icône	Option	Description
	Table suivante	Si vous êtes revenu en arrière dans la séquence de tables affichées en utilisant l'option <i>Table précédente</i> , vous pouvez sélectionner cette option pour avancer dans la séquence des tables MGL affichées. Si vous avez atteint la dernière table (la plus récente) dans la séquence, cette option est désactivée.
	Sauter à la rangée	Cliquez sur ce bouton pour sauter à une rangée spécifique dans la table couramment affichée. La fenêtre suivante s'affiche:



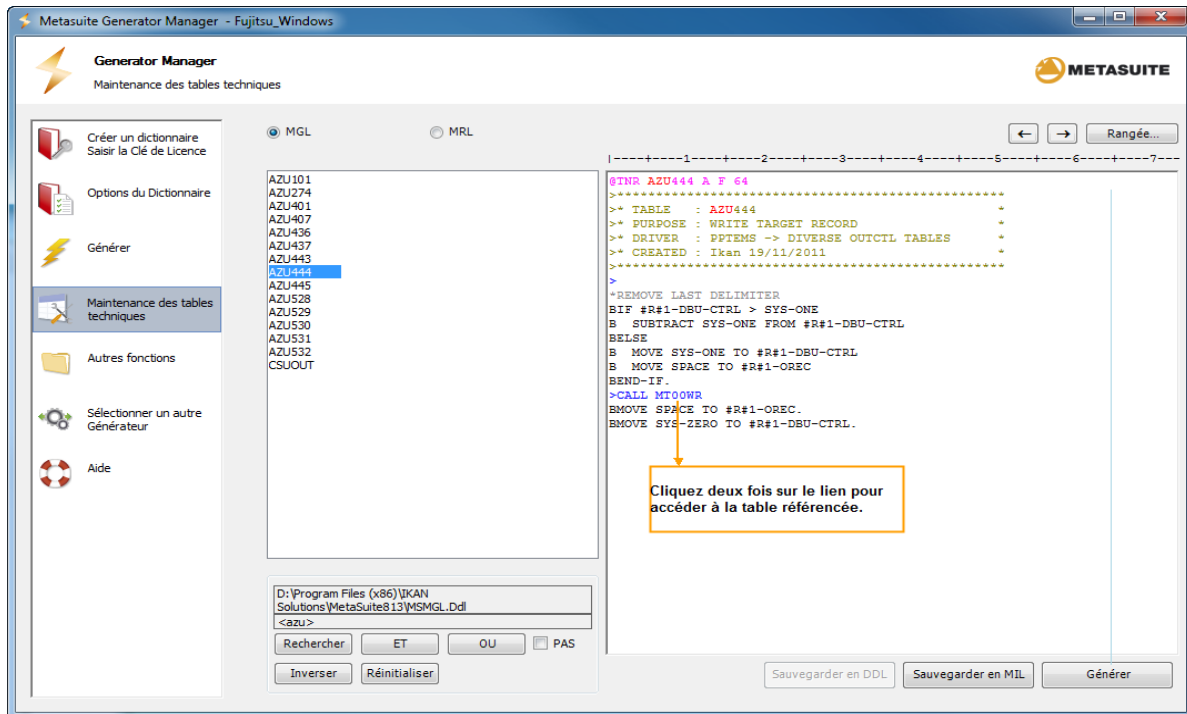
Saisissez le numéro de rangée requise et cliquez sur le bouton OK.
Résultat:

La rangée sélectionnée est surlignée dans le Fichier:

```
@TNR AV0003 A F 60
* SYSTEM LST PARMS
A01 SYS-LST-PARMS.
B02 SYS-PROGRAM PIC X(8) VALUE "#1".
B02 SYS-VERSION PIC X(4) VALUE "#2".
B02 SYS-WRITTEN PIC X(14) VALUE "#3".
B02 SYS-LST-DATE PIC S9(8) BINARY VALUE 0.
B02 SYS-LST-FUNCT PIC S9(4) BINARY VALUE 0.
B02 SYS-LST-CC PIC S9(4) BINARY VALUE 0.
B02 SYS-LST-MSG PIC S9(4) BINARY VALUE 0.
B02 SYS-LST-RECL PIC S9(4) BINARY VALUE 0.
B02 SYS-LST-SNAP PIC S9(8) BINARY.
B02 SYS-LST-SNAP-LIMIT PIC S9(8) BINARY.
B02 SYS-LST-EFLD.
C03 SYS-LST-EFIELD PIC X(42).
C03 SYS-LST-EFTYPE PIC X(8). → ligne 15
C03 SYS-LST-EFLEN PIC 9(8).
C03 SYS-LST-EFDORD PIC 9(2).
C03 SYS-LST-EFPOS PIC 9(8).
```

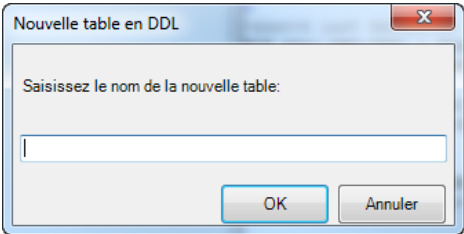
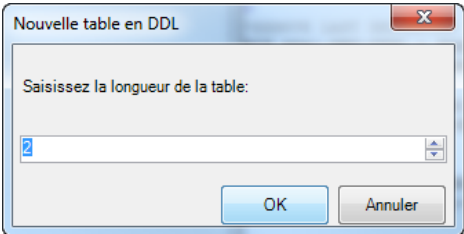
Note: Les rangées commençant par @ sont ignorées.

5. Si une table se réfère à une autre table, vous pouvez cliquer deux fois sur le nom de la table pour l'accéder directement.



6. Modifiez le code si nécessaire.

Note: Si vous cliquez avec le bouton droit sur le contenu de la table, un menu déroulant s'affichera contenant plusieurs commandes telles que Copier, Coller, Imprimer, ...

Option	Description
Nouveau	Cliquez avec le bouton droit sur le contenu de la table et sélectionnez <i>Nouveau</i> dans le menu. La fenêtre suivante s'affiche:  Saisissez le nom de la nouvelle table. Il doit contenir exactement 6 caractères. Ensuite cliquez sur le bouton OK. La fenêtre suivante s'affiche:  Saisissez le nombre de lignes attendues et cliquez sur le bouton OK. La table est créée. Maintenant, vous pouvez saisir la codification nécessaire et sauvegarder la table vers un fichier DDL en utilisant l'option <i>Sauvegarder en DDL</i>.

7. Une fois que vous avez fait les modifications nécessaires, vous pouvez utiliser une des options suivantes: *Sauvegarder en DDL*, *Sauvegarder en MIL* ou *Générer*

Note: Les options suivantes sont disponibles dans le menu.

- Sauvegarder en DDL

Cliquez sur l'option *Sauvegarder en DDL* pour sauvegarder la version actuellement affichée d'une table vers le fichier MSMGL.DDL.

Cela est nécessaire:

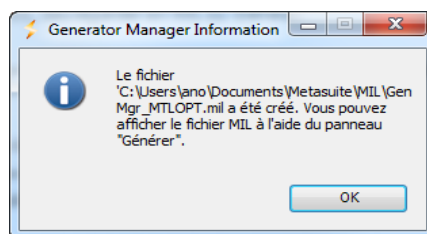
- après avoir modifié un fichier existant
- après avoir créé un nouveau fichier

La nouvelle version de la table est sauvegardée dans le fichier **MSMGL.DLL** fichier.

- Sauvegarder en MIL

Cliquez sur le bouton *Sauvegarder en MIL* pour sauvegarder les modifications d'une table vers un nouveau fichier MIL dans le répertoire standard des fichiers MIL.

La fenêtre suivante s'affiche:



Le format du nom du fichier est comme suit:

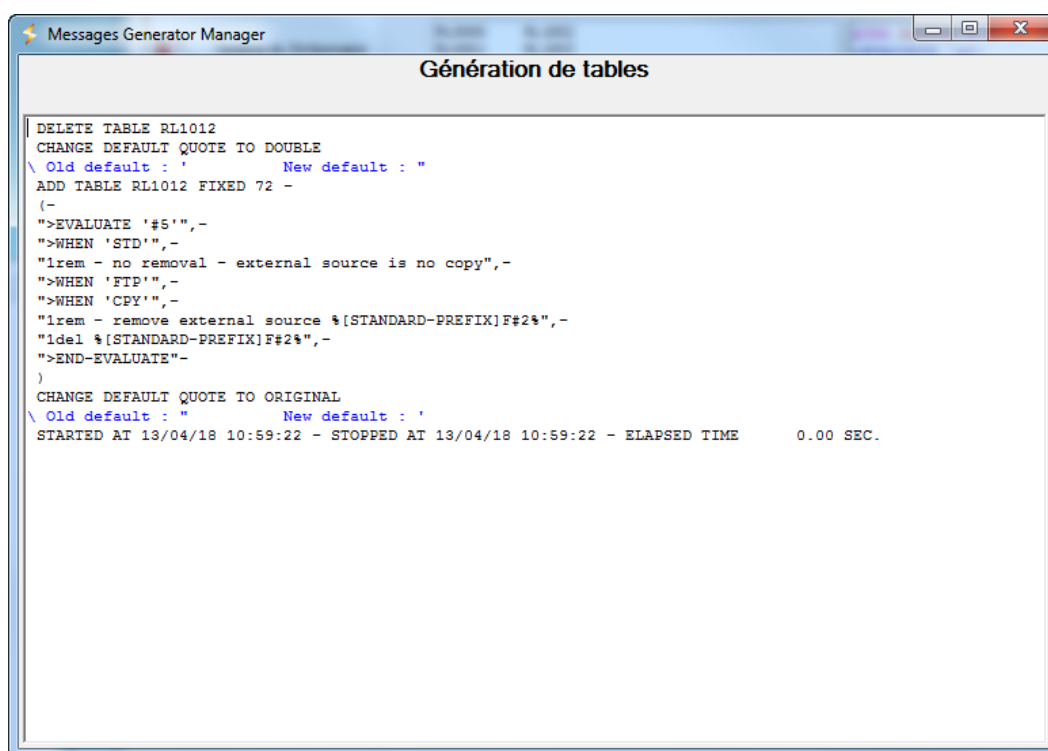
GenMgr_TableName.MIL

Implémentez le Fichier MIL pour rendre les changements effectifs. Voir [L'écran "Générer pour MIL"](#) on page 19.

- Générer

Cliquez sur le bouton *Générer* pour rendre les changements immédiatement effectifs dans le Dictionnaire du Générateur.

La fenêtre suivante s'affiche:



11.2. Tables MGL: Description détaillée

Si vous avez terminé la création d'un Modèle MetaMap, vous devez générer le programme de code source COBOL correspondant (MGL). Pour la génération des instructions COBOL liées au traitement du fichier, le Générateur utilise les tables MGL.

Il existe 4 types de tables MGL:

- Tables Système
- Tables de Contrôle des Fichiers
- Tables de Code-Contrôle

- Tables de code COBOL

Le tableau suivant décrit ces types:

Type de table MGL	Description
Tables Système	Le Générateur utilise les Tables Système à des fins différentes. Exemples: • AU0001: contient tous les messages du Générateur • AU0002: contient tous les mots réservés • AU0006: contient tous les noms des variables Système • MTLOPT: contient les options MTL • MTLINI: contient les paramètres initiaux pour certaines variables MTL
Tables de Contrôle des Fichiers	Les Tables de Contrôle des Fichiers sont encore utilisées par certains types de données au lieu des Tables de Code-Contrôle. Elles contiennent une seule ligne constituée de 32 caractères. Vu que leur utilisation devient obsolète, elles ne sont pas décrites en détail. Exemple: AU0030, AU0031
Tables de Code-Contrôle	Des Tables de Code-Contrôle sont disponibles pour chaque type de Fichier supporté par le système. Chacune de ces Tables de Code-Contrôle comprend un nombre fixe d'entrée numérotées qui se rapportent à un point spécifique dans le processus de génération d'un programme COBOL. Par exemple, il existe une entrée qui détermine quel code sera généré pour lire un Fichier. Nous vous recommandons vivement de NE PAS éditer les Tables de Code-Contrôle par défaut. Si un Fichier ou une Base de données spécifique requiert des paramètres particuliers, il vaut mieux créer une nouvelle Table de Code-Contrôle spécifique pour ce Fichier ou pour cette Base de données. Il existe deux catégories de Tables de Code-Contrôle: • Les Tables de Code-Contrôle Source (également appelées des Tables de Code-Contrôle INPUT ou GEN). Voir Tables de Code-Contrôle Source on page 67. • Les Tables de Code-Contrôle Cible (également appelées des Tables de Code-Contrôle OUTPUT). Voir Tables de Code-Contrôle Cible on page 71.
Tables de code COBOL	Les Tables de code COBOL du Générateur sont nommées des collections de squelettes de code COBOL. Chaque table de code peut contenir une ou plusieurs lignes de code COBOL. Les tables de code sont sauvegardées dans la bibliothèque du Générateur (c.-à-d., le dossier MGL). Il est possible de définir des paramètres dans le code COBOL. Ils sont indiqués par un '#' suivi d'un seul chiffre ou d'un seul caractère. À cause d'une convention générale utilisée par le système, les Champs de Fichiers sont formatés comme F#1-xxxx dans une table de code. Pendant le processus de génération du code actuel, le '#1' est remplacé par le numéro relatif du Fichier. Voir Tables de code COBOL on page 74.

11.3. Tables de Code-Contrôle

Le Générateur MetaSuite utilise les Tables de Code-Contrôle pour contrôler la génération des instructions COBOL liées au traitement d'un Fichier dans un programme COBOL généré par MetaSuite.

Pour chaque type de Fichier supporté par le système, une Table de Code-Contrôle est fournie avec MetaSuite. Chacune de ces Tables de Code-Contrôle comprend un nombre maximal de 54 entrées numérotées qui se rapportent à un point spécifique dans le processus de génération d'un programme COBOL. Par exemple, il existe une entrée qui détermine quel code sera généré pour lire un Fichier.

Il existe deux types de Tables de Code-Contrôle:

1. Les Tables de Code-Contrôle Source, contenant maximum 54 entrées numérotées, également appelées des Tables de contrôle GEN ou INPUT.
2. Les Tables de Code-Contrôle Cible, contenant maximum 32 entrées numérotées, également appelées des Tables de contrôle OUTPUT.

Ces Tables de Code-Contrôle standard sont sauvegardées pendant l'installation du Dictionnaire du Générateur MetaSuite. Chaque fois qu'un Utilisateur ajoute (avec la commande ADD FILE) une définition de Fichier au Dictionnaire, sans spécifier une Table de Code-Contrôle dans la commande ADD FILE, le système utilisera par défaut la Table de Code-Contrôle standard installée pour le type de Fichier spécifié dans la définition. Une Table de Code-Contrôle standard suppose ce qui suit.

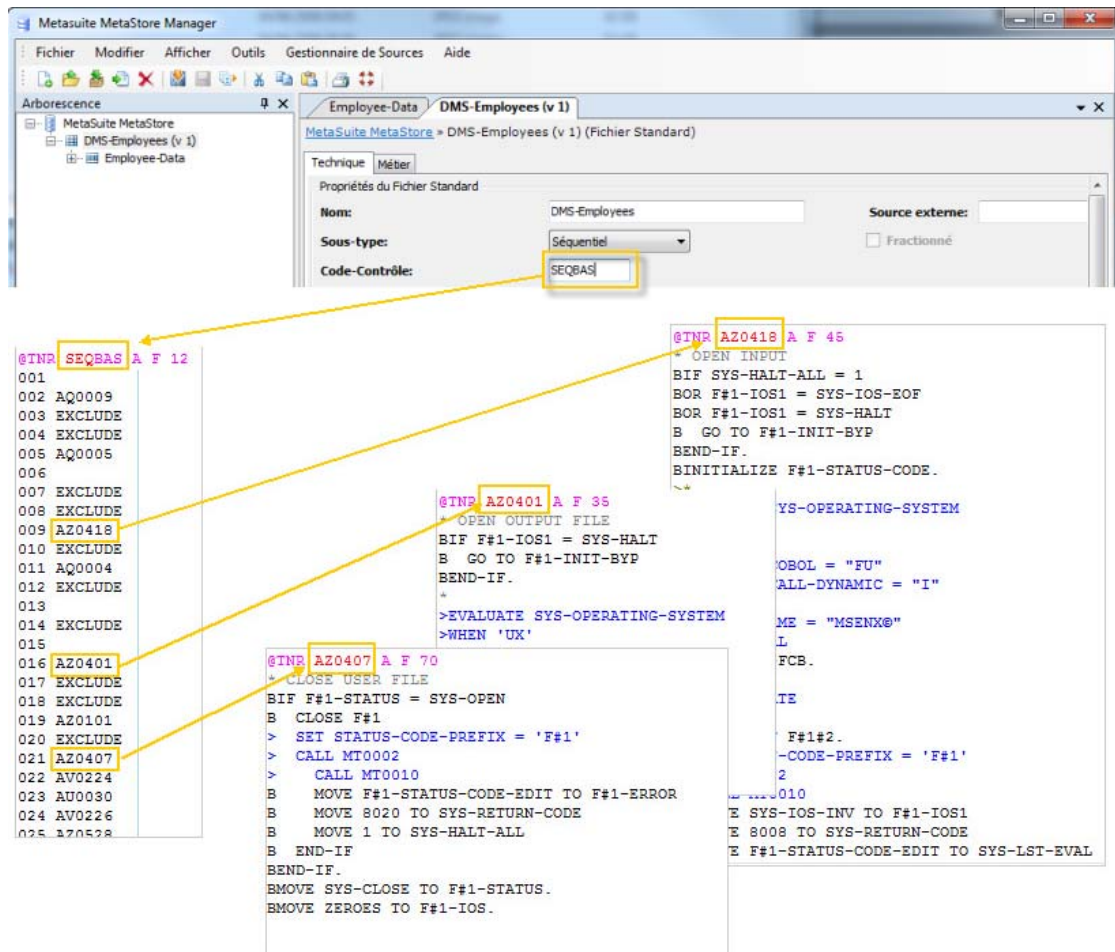
- Le Fichier peut être ouvert, lu, traité et fermé en utilisant les termes COBOL standard.
- Le(s) Enregistrement(s) de Fichier peuvent être décrits en utilisant les instructions COBOL de définition de données standard.

Pour la majorité des Fichiers, les Tables de Code-Contrôle standard fournissent toutes les informations nécessaires pour générer le code COBOL pour lire et traiter les Fichiers.

Cependant, il y a des Fichiers pour lesquels les Tables de Code-Contrôle standard ne fonctionneront pas parce qu'elles ne répondent pas aux propositions décrites ci-dessus (par exemple, un fichier compressé). En général, il existe deux types pour de tels fichiers:

1. Un Fichier qui ne peut pas être accédé en utilisant les termes COBOL standard. Un sous-programme ou un ensemble de sous-programmes écrits par l'Utilisateur doit être appelé pour exécuter chaque E/S et pour exécuter toute décompression ou reformatage requis.
2. Un Fichier peut être ouvert, lu et fermé en utilisant les termes COBOL standard, mais ses Enregistrements doivent être décompressés, ou formatés différemment, par un sous-programme fourni par l'Utilisateur immédiatement après la lecture du Fichier et avant tout traitement des données de l'Enregistrement.

Le tableau suivant explique comment les Tables de Code-Contrôle sont organisées.



Format

Une table de Code-Contrôle Source standard est une table de taille fixe avec 54 entrées. Chaque entrée est numérotée, de 1 à 53, et mise en page comme suit:

NNN XXXXXX

Où **NNN** est un nombre à 3 chiffres (y compris les zéros en tête), de 1 à 54.

XXXXXX est un des types de spécification suivants:

- Le nom d'une table de code COBOL, stockée dans le Dictionnaire du Générateur, qui est généré pour le Fichier au point correspondant dans le processus de génération COBOL.
- L'option EXCLUDE indique au Générateur MetaSuite qu'aucune table de code ne doit être générée pour le Fichier au point de traitement correspondant.
- Des espaces indiquent au Générateur MetaSuite de suivre la génération de table de code par défaut.

Note: Cette spécification ne s'applique qu'aux Tables de Code-Contrôle standard. Vous ne pouvez pas spécifier des espaces pour une entrée de Table de Code-Contrôle dans une table écrite par l'Utilisateur.

Tables de Code-Contrôle Source

Les Tables de Code-Contrôle Source du Générateur MetaSuite sont des Tables de Code-Contrôle utilisées pour traiter les Fichiers Source.

Les Tables de Code-Contrôle pour les Fichiers Source sont définies en combinaison avec l'option CODE-CONTROL de la commande ADD FILE, décrite dans le chapitre *Commandes du langage de définition* dans le Guide de l'Utilisateur de MetaStore Manager.

Les Tables de Contrôle des Fichiers Source sont:

AU0000	Fichier DUMMY
AU0010	Fichier plat SEQUENTIEL (IBM QSAM)
AU0023	Fichier INDEX
AU0024	Fichier RANDOM
AU0025	Fichier IBM VSAM KSDS
AU0026	Fichier IBM VSAM ESDS
AU0027	Fichier IMS
AU0028	Fichier IDMS
AU0034	Fichier TOTAL
AU0035	Fichier ADABAS
AU0039	DATAACOM/DB
AU0046	Fichier DBS
D20050	DB2 WINDOWS
D2I050	DB2 WINDOWS RESTART
D40050	DB2 AS4
D60050	DB2 RS6000
DM0050	DB2 Z/OS
DMI050	DB2 Z/OS RESTART
DV0050	DB2 DOS/VSE
FC0050	ODBC WINDOWS Fujitsu
IF0050	INFORMIX
IG0050	INGRES
IM0010	Fichier GSAM
MQSINP	MQ-SERIES INPUT FILE
OR0050	ORACLE
PARINP	Entrée de fichier de paramètres
RD0050	RDBMS
SE0050	SESAM (OSD/BS2000)
SEQBAS	Fichier plat délimité (séparé par des virgules)
SQ0050	SQLSERVER
SY0050	SYBASE

TD0050	TERADATA
XMLINP	(XML (primitif - sans contrôle de chemin)
XPCIFX	XML avec contrôle de chemin, format de données fixe

Le tableau suivant décrit les entrées dans une Table de Code-Contrôle Source standard:

Entrée	Fonction	Génère
001	SELECT	Instructions SELECT/ASSIGN
002	FD	Instructions FD
003	FDRECS	Mise en page d'Enregistrement FD
004	RECORD-PREP	Code de préparation d'Enregistrement unique
005	WS-MISC	Zone de Travail/Stockage divers
006	WS-RECS	Mise en page d'Enregistrement de Travail/Stockage
007	LS-MISC	Zone particulière de LINKAGE-SECTION
008	LS-RECS	Définition du RECORD en LINKAGE-SECTION
009	OPEN-INPUT	Code d'ouverture de fichier en entrée
010	POST-OPEN-IN	Post code d'ouverture de fichier en entrée
011	READ	Code de lecture séquentielle
012	READ-KEYED	Code de lecture avec Clé
013	POST-READ	Post Code de lecture
014	SET-SYSREC	Spécifie le code SYSREC
015	TASK-INIT	Code pour démarrer la tâche (après initialisation du programme)
016	OPEN-OUTPUT	Code d'ouverture de fichier en sortie
017	POST-OPEN-OUT	Post Code d'ouverture de fichier en sortie
018	PRE-WRITE	Code de Pré-écriture d'enregistrement
019	WRITE	Code d'écriture séquentielle
020	WRITE-KEYED	Code d'écriture avec clé
021	CLOSE	Code de fermeture de Fichier
022	BUF-TABHEAD	Code de contrôle de la mémoire tampon (pour utilisation interne)
023	FILE-CONTROL	Table de contrôle de Fichier

Entrée	Fonction	Génère
024	BUF-TABEND	Code de contrôle de la mémoire tampon (pour utilisation interne)
025	BUF-READCTRL	Code de contrôle de la mémoire tampon (pour utilisation interne)
026	READ-IX-KEYED	Code de clé de lecture indexée
027	BUF-SIMPLOAD	Code de contrôle de la mémoire tampon (pour utilisation interne)
028	BUF-OCCLOAD	Code de contrôle de la mémoire tampon (pour utilisation interne)
029	BUF-LOADED	Code de contrôle de la mémoire tampon (pour utilisation interne)
030	BUF-EOF	Code de contrôle de la mémoire tampon (pour utilisation interne)
031	ENTRYNAME	Nom de point d'entrée
032	ENTRYPARM	Paramètres de point d'entrée
033	GETNAM	Configuration d'un Enregistrement pour un accès avec clé
034	GETKEY	Configuration d'une Clé pour un accès avec clé
035	BUF-TABENT1	Code de contrôle de la mémoire tampon (pour utilisation interne)
036	BUF-TABENT2	Code de contrôle de la mémoire tampon (pour utilisation interne)
037	BUF-TABENT3	Code de contrôle de la mémoire tampon (pour utilisation interne)
038	BUF-TABENT4	Code de contrôle de la mémoire tampon (pour utilisation interne)
039	BUF-TABENT5	Code de contrôle de la mémoire tampon (pour utilisation interne)
040	BUF-TABENT6	Code de contrôle de la mémoire tampon (pour utilisation interne)
041	START	Code de démarrage de Fichier
042	FLIST-NAME	Code pour une liste formatée (usage interne)
043	FLIST-NAME1	Code pour une liste formatée (usage interne)
044	FLIST-HEADER	Code pour une liste formatée (usage interne)
045	FLIST-END	Code pour une liste formatée (usage interne)
046	TASK-START	Code pour démarrer la tâche (après initialisation du programme)

Entrée	Fonction	Génère
047	TASK-STOP	Code d'arrêt de tâche
048	DUPL-READ	Code pour une Clé de lecture en double
049	FD-DUMMY-REC	Squelette d'Enregistrement FD
050	WS-DUMMY-REC	Squelette d'Enregistrement de travail/ stockage
051	LS-DUMMY-REC	Squelette d'Enregistrement de LINKAGE-SECTION
052	PD-MISC	PROCEDURE-DIVISION particulière
053	-	Réservé pour usage ultérieur.
054	-	Réservé pour usage ultérieur.

Tables de Code-Contrôle Cible

Les Tables de Code-Contrôle Cible du Générateur MetaSuite, ou les Tables de Contrôle de sortie, sont des Tables de Code-Contrôle utilisées pour traiter les Fichiers Cibles.

Elles ont la même structure que les Tables de Code-Contrôle pour les Fichiers Source, mais le nombre d'entrées et leur signification sont différentes.

Les Tables de Code-Contrôle pour les Fichiers Cible sont définies en combinaison avec l'option OUTPUT-CONTROL de la commande TARGETFILE, décrite dans le chapitre *Commandes du langage de définition* dans le Guide de l'Utilisateur de MetaStore Manager.

Les Tables de Contrôle des Fichiers Cible sont:

AU0041	DIF
AU0042	Fichiers SYLK
AU0043	Valeur séparée par des virgules (type délimité standard)
AU0044	PRN
AU0045	SAS
AU0051	Fichier plat variable, Lignes séquentielles (IBM QSAM)
AU0052	ORACLE
AU0054	SYBASE
AU0055	SQLSERVER
AU0056	BRS/Abacus
AU0057	Fichiers Cible fixes séquentiels (IBM QSAM)
AU0058	IMS GSAM
AU0061	Lignes séquentielles contrôlées - variable
AU0067	Lignes séquentielles contrôlées - fixe
AU0068	Lignes séquentielles contrôlées -IMSGSAM
AU0151	Fichier plat variable, Enregistrement séquentiel

AU0157	Fichier plat fixe, Enregistrement séquentiel
AU0161	Enregistrements séquentiels contrôlés - variable
AU00167	Enregistrements séquentiels contrôlés - fixe
CSXOUT	Séquentiel contrôlé - XML
HTMOUT	Fichier HTML
PAROUT	Fichier de paramètres - une seule sortie d'Enregistrement
SQLOUT	Table SQL comme Fichier Cible (réservé - non encore supporté)
MQSOUT	Fichier de sortie MQ-Series (réservé - non encore supporté)
TR0057	TERADATA fixe séquentiel
XMLOUT	Type XML (Fichier XML simple)
XPCOUT	XML avec contrôle de chemin (réservé - non encore supporté)

Le tableau suivant décrit les entrées dans une Table de Code-Contrôle Cible standard:

Entrée	Fonction	Génère
001	Contient des modifications	Des modifications d'En-tête, de Fin, d'Enregistrement de données et de Titre.
002	Transfert/Rapport FD	Description de Fichier, contenant l'En-tête et les descriptions de détails des Enregistrements
003	Transfert/Rapport SELECT	Instruction SELECT
004	Transfert/Rapport OPEN	Code d'ouverture de fichier (OPEN output)
005	Transfert/Rapport WRT-HDR	Écriture de l'en-tête de l'Enregistrement
006	Transfert/Rapport WRT-FLD-ALPH	Écriture de Champ alphanumérique
007	Transfert/Rapport WRT-FLD-NUM	Écriture de Champ numérique
008	Transfert/Rapport WRT-FLD-DATE	Écriture de Champ de type date
009	Transfert/Rapport WRT-FLD-GEN	Écriture de Champs en général, pour la plupart des types d'Enregistrements basés sur SQL
010	Transfert/Rapport WRT-TRLR	Écriture d'un Enregistrement de fin
011	Transfert/Rapport CLOSE	Code de fermeture de fichier (CLOSE)
012	Détails FD	Description de Fichier pour la sortie de Détails, pas pour les Rapports
013	Détails SELECT	Instruction SELECT pour un Enregistrement de Détails
014	Détails HDR-STRUCT	Définition de l'En-tête du Champ
015	Détails FLD-STRC-ALPH	Définition d'un Champ alphanumérique

Entrée	Fonction	Génère
016	Détails FLD-STRC-NUM	Définition d'un Champ numérique
017	Détails FLD-STRC-DATE	Définition d'un Champ de type date
018	Détails FLD-STRC-GEN	Définition d'un Champ général
019	Détails FLD-TLRL	Définition d'un Champ de fin
020	Détails OPEN	Définition d'ouverture (OPEN output) d'un Fichier Cible
021	Détails WRT-HDR	Écriture d'un En-tête
022	Détails MOV-ALPH	Instructions de déplacement pour des Champs alphanumériques
023	Détails MOV-NUM	Instructions de déplacement pour des Champs numériques
024	Détails MOV-DATE	Instructions de déplacement pour des dates
025	Détails MOV-GEN	Instructions de déplacement pour des Champs généraux
026	Détails WRT-DATAREC	Écriture des détails de l'Enregistrement
027	Détails WRT-TRLR	Écriture d'un Enregistrement de fin
028	Détails du CLOSE	Fermeture (CLOSE) du Fichier Cible
029	WS-MISC	Zone particulière de Travail/Stockage
030	LS-MISC	Zone particulière de LINKAGE-SECTION
031	PD-MISC	PROCEDURE-DIVISION particulière
032	TDF-WRT-DET	Écriture des détails - initialisation

Tables de Code-Contrôle de l'Utilisateur

Les Tables de Code-Contrôle de l'Utilisateur du Générateur MetaSuite sont des Tables de Code-Contrôle développées par l'Utilisateur.

Le format de chaque entrée dans une Table de Code-Contrôle définie par l'Utilisateur est le même que celui décrit ci-dessus pour les Tables de Code-Contrôle standard. Cependant, une table définie par l'Utilisateur n'inclut que les entrées qui diffèrent de celles dans la table standard.

Par exemple, supposons qu'un Fichier requiert un traitement spécial pour un Enregistrement de saisie immédiatement après que cet Enregistrement soit lu. Vous pourriez définir une Table de Code-Contrôle qui écrase uniquement l'entrée POST-READ dans la table standard. Les commandes d'entrée d'administration système pourraient ressembler à:

```
CHANGE DEFAULT QUOTE
ADD TABLE MYTABL FIXED 12 -
("013 POSTRD")
CHANGE DEFAULT QUOTE
```

La commande ADD FILE pour MetaSuite MetaStore peut être codifiée comme suit:

```
ADD FILE MYFILE TYPE SEQUENTIAL FIXED 120 -
CODE-CONTROL MYTABL
```

Vu qu'il s'agit d'un Fichier séquentiel, le Générateur MetaSuite utilisera la Table de Code-Contrôle standard pour un Fichier séquentiel, en remplaçant uniquement l'entrée 13, tel que spécifié dans la table MYTABL. Pour plus d'informations concernant la création de vos propres Tables de Code-Contrôle, se référer à la section [Créer des Tables de Code-Contrôle personnalisées](#) (page 76)

11.4. Tables de code COBOL

Les Tables de code COBOL MetaSuite sont nommées des collections de squelettes de code COBOL.. Chaque table de code peut contenir une ou plusieurs lignes de code COBOL. Les Tables de Code sont maintenues dans le Dictionnaire du Générateur en utilisant les commandes ADD/DELETE TABLE décrites dans ce document.

Le nom de la Table de Code doit être unique dans le Dictionnaire du Générateur.

Dans une Table de Code, l'instruction COBOL doit respecter les conventions de codification COBOL. Dans la plupart des cas, vous pouvez commencer par vérifier la Table de Code standard qui serait générée pour le Fichier.

Notez que la codification prototype peut être paramétrée. Un paramètre est indiqué par un '#' suivi d'un seul numéro ou caractère. À cause d'une convention générale utilisée par le système les Champs de Fichiers sont formatés comme F#1-xxxx dans une table de code. Pendant le processus de génération de code actuel, le '#1' est remplacé par le numéro relatif du Fichier. Par exemple, le nom F#1-STATUS dans une Table de Code s'afficherait comme F01-STATUS dans le programme s'il est généré pour le premier Fichier nommé dans le programme Source COBOL.

Le paramétrage des Tables de code est déterminée par la codification système interne du Générateur. Si vous avez des questions concernant le paramétrage, contactez l'équipe de Support MetaSuite.

Pour plus d'informations concernant la création de vos propres Tables de Code COBOL, se référer à la section [Créer des Tables de code COBOL personnalisées](#) (page 76)

11.5. Exemples de table de code

Exemple 1 - Table de Code pour appeler des Sous-programmes E/S

Dans cet exemple nous supposons que vous utilisez un Fichier qui doit être ouvert, lu et fermé en utilisant un de vos propres sous-programmes E/S. Dans ce cas, vous devez créer une Table de Code-Contrôle Utilisateur pour le Fichier qui écrase les entrées de la Table de Code-Contrôle.

Les commandes MIL pour ajouter des nouvelles Tables de Code-Contrôle et des nouvelles Tables de Code sont les suivantes:

```
CHANGE DEFAULT QUOTE
ADD TABLE USRTBL FIXED 12 -
("001 EXCLUDE", - (do not generate a SELECT)
 "002 EXCLUDE" , - (do not generate a FD)
 "003 EXCLUDE" , - (do not generate a FD record layout)
 "005 USRWSM" , - (alternate Working-Storage)
 "006" , - (generate record layout in W-S)
 "009 USROPN " , - (call user routine to open file)
 "011 USRRED" , - (call user routine to read file)
 "021 USRCLS") (call user routine to close)

ADD TABLE USRWSM FIXED 56 -
("* USRWSM FILE WORKING-STORAGE AREA", -
 "A01 F#1 - WS-MISC.", -
 "B02 F#1 - STATUS-CODE.", -
```

```
"C03  F#1 - STATUS-CODE1      PIC X VALUE '0'.", -
"C03  F#1 - STATUS-CODE2      PIC X VALUE '0'.", -
"A01  F#1 - USER-PARMS.", -
"B02  F#1 - USER-CODE         PIC  X.)"
```

(Les Champs F#1-WS-MISC, F#1-STATUS-CODE, F#1-STATUS-CODE1 et F#1-STATUS-CODE2 sont des Champs de statut requis pour un Fichier séquentiel.)

```
ADD TABLE USROPN FIXED 56
(*  USROPN-USER FILE OPEN", -
"BMOVE '0' TO F#1-USER-CODE.", -
"BCALL 'USRROUTIN' USING F#1-USER-CODE", -
"B          F#1-WSREC.)"
```

(Le nom F#1-WSREC est toujours généré par le Générateur de MetaSuite pour toute structure d'Enregistrement généré dans "Working-Storage".)

```
ADD TABLE USRRED FIXED 56 -
(*  USRRED-USER FILE READ", -
"BMOVE 'R' TO F#1-USER-CODE.", -
"BCALL 'USRROUTIN' USING F#1-USER-CODE", -
"B          F#1-WSREC.", -
"BIF F#1-USER-CODE EQUAL 'E', -
"CMOVE SYS-IOS-EOF- TO F#1-IOS1.)"
```

```
ADD TABLE USRCLS FIXED 56 -
(*  USRCLS-USER FILE CLOSE", -
"BMOVE 'C' TO F#1-USER-CODE.", -
"BCALL 'USRROUTIN' USING F#1-USER-CODE", -
"B          F#1-WSREC.)"
```

```
CHANGE DEFAULT QUOTE
```

Exemple 2 - Table de Code pour appeler des Sous-programmes de reformatage

Dans cet exemple nous supposons que vous avez un Fichier qui peut être ouvert, lu et fermé en utilisant des termes COBOL standard. Cependant, l'Enregistrement doit être décompressé avant que le programme ne puisse en référencer des Champs. Dans ce cas, vous devez créer une Table de Code-Contrôle Utilisateur pour le Fichier qui écrase les entrées de la Table de Code-Contrôle pour générer une structure d'Enregistrement FD, en générant une structure d'Enregistrement "Working-Storage" (pour l'Enregistrement décompressé) et en effectuant un traitement après lecture.

Les commandes d'administration de système pour ajouter des nouvelles Tables de Code-Contrôle et des nouvelles Tables de Code sont les suivantes:

```
CHANGE DEFAULT QUOTE
ADD TABLE USRTBL FIXED 12-
("003  USERFD", - (generate a user FD record)
"006", -          (generate record layout in W-S)
"013  POSTRD") ( call decompress routine)
```

```
ADD TABLE USERFD FIXED 56 -
(*  USERFD FILE FD RECORD AREA", -
"A01  F#1-FDREC      PIC X (300).")
```

```
ADD TABLE POSTRD FIXED 56-
(*  POSTRD-POST READ COD", -
"BMOVE F#1-FDREC TO F#1-WSREC.", -
"BCALL 'USRROUTIN' USING F#1-FDREC", -
"B  F#1-WSREC.)"
```

(Le nom F#1-WSREC est toujours généré par le Générateur de MetaSuite pour toute structure d'Enregistrement généré dans "Working-Storage".)

11.6. Créer des Tables de Code-Contrôle personnalisées

1. Accédez à l'écran *Maintenance des tables techniques*.
Voir [L'écran de Maintenance des tables techniques MGL](#) on page 59.
2. Sélectionnez une Table de Code-Contrôle standard dont le contenu ressemble à la table personnalisée requise.
3. Remplacez le nom de la table à la première ligne.
Choisissez un nom qui indique clairement le but de la Table de Code-Contrôle personnalisée.
4. Si nécessaire, modifiez les paramètres.
Le format de chaque entrée dans une Table de Code-Contrôle personnalisée est identique au format tel que décrit ci-dessus pour une Table de Code-Contrôle standard. Cependant, une table définie par l'Utilisateur n'inclut que les entrées qui diffèrent de celles dans la table standard.
5. Cliquez sur le bouton *Sauvegarder en DDL*.
La nouvelle table sera affichée dans la liste des tables MGL disponibles.
6. Cliquez sur le bouton *Générer* pour rendre la nouvelle table disponible dans le Dictionnaire du Générateur.
7. Dans MetaStore Manager, vous pouvez assigner la Table de Code-Contrôle à la définition de la Source de données requise.

11.7. Créer des Tables de code COBOL personnalisées

1. Accédez à l'écran *Maintenance des tables techniques*.
Voir [L'écran de Maintenance des tables techniques MGL](#) on page 59.
2. Sélectionnez une Table de Code COBOL standard dont le contenu ressemble à la table personnalisée requise.
3. Remplacez le nom de la table à la première ligne.
Choisissez un nom qui indique clairement le but de la Table de Code COBOL personnalisée.
4. Si nécessaire, modifiez les paramètres.
Une Table de Code COBOL définie par l'Utilisateur contient des commentaires, des commandes COBOL et des commandes MTL (Langage de modèle MetaSuite).
Aperçu de la syntaxe

Caractère de départ	La codification générée démarre dans la colonne	Description
1	1	Options de compilateur COBOL ou code JCL
*	7	Commentaires COBOL
/	7	Saut de page COBOL
A	8	Codification COBOL marge-A
B	12	Codification COBOL marge-B
C	16	Codification COBOL colonne 16
D ou \$	12	Codification "dump" (uniquement générée en mode de débogage)
-	-	Codification inactive
>	-	Commande de Langage de modèle MetaSuite (page 99)
@	-	Instructions DDL lors de la création d'un nouveau Dictionnaire

5. Cliquez sur le bouton *Sauvegarder en DDL*.
La nouvelle table sera affichée dans la liste des tables MGL disponibles.
6. Cliquez sur le bouton *Générer* pour rendre la nouvelle table disponible dans le Dictionnaire du Générateur.

Maintenance des tables techniques

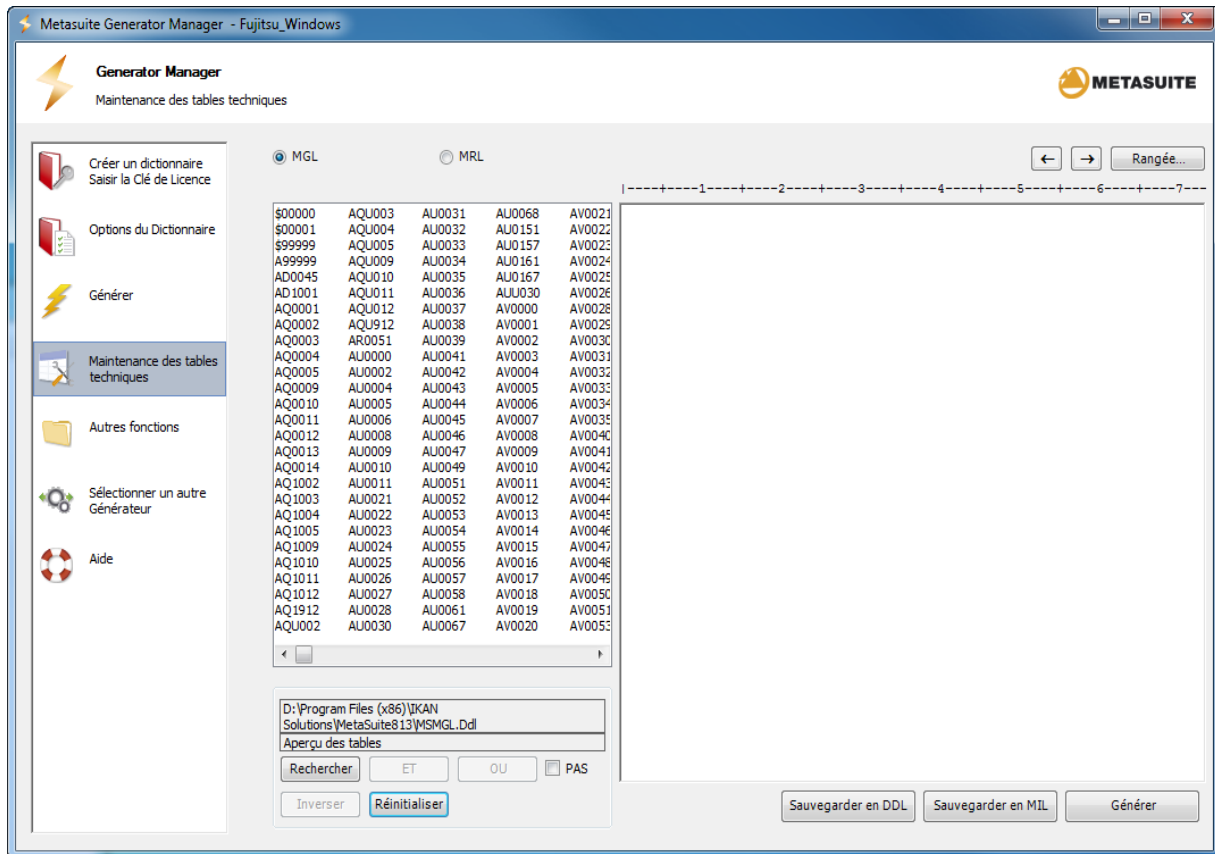
- Maintenance des tables MRL

L'abréviation MRL signifie MetaSuite Run Language (Langage d'exécution MetaSuite). Le Fichier MSMRL.DDL (placé dans le dossier DDL du type de Générateur de l'Installation MetaSuite) est un des trois Fichiers chargés pour le Dictionnaire du Générateur. Le Fichier MSMRL.DDL contient des tables MRL constituées de petits morceaux de scripts. Ils décrivent comment les scripts MRL seront générés quand un Modèle MetaMap est généré.

Note: Il est logique de créer un nouveau Dictionnaire après avoir édité les tables MRL (uniquement pour les Utilisateurs expérimentés!).

12.1. L'écran de Maintenance des tables techniques MTL

1. Cliquez sur le bouton *Maintenance des tables techniques* dans la colonne de gauche.
L'écran suivant s'affiche:



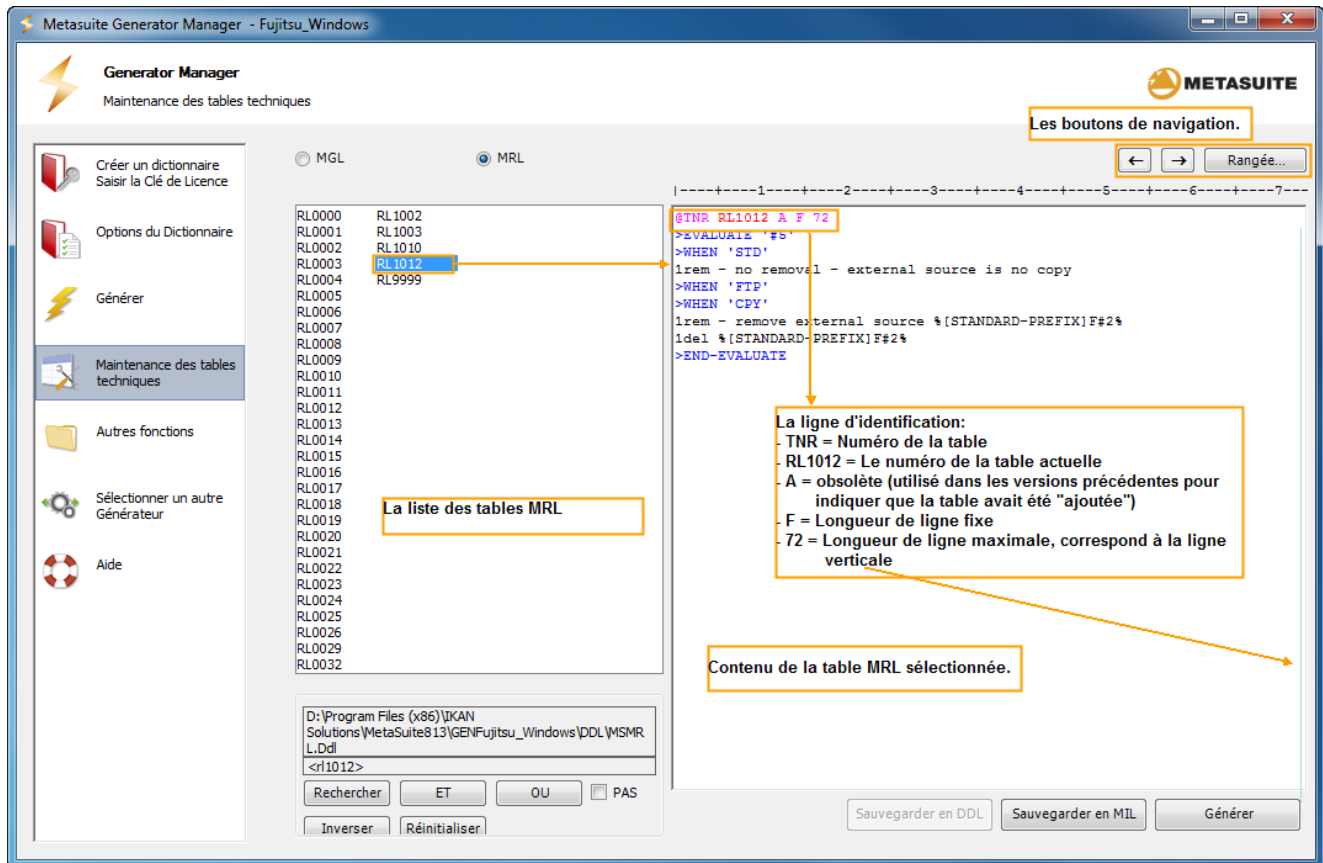
Pour une explication des parties communes, se référer au chapitre [Maintenance des tables techniques - Aperçu](#) (page 54)

2. Sélectionnez l'option de format *MRL*.
Les tables MRL disponibles s'affichent.

Note: Si vous voulez limiter la liste des tables MRL affichées, vous pouvez définir des chaînes de caractères recherchés comme illustré dans le chapitre [Maintenance des tables techniques - Aperçu](#) (page 54).

3. Cliquez sur une table MRL pour en afficher son contenu à droite.

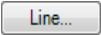
L'écran suivant s'affiche:

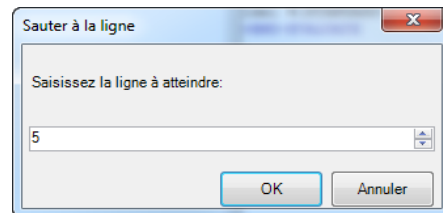


4. Si nécessaire, utilisez les boutons de navigation.

Les boutons suivants sont disponibles:

Icône	Option	Description
	Table précédente	Si vous avez affiché une séquence de tables, vous pouvez sélectionner cette option pour revenir en arrière dans la séquence et pour réafficher la table MRL précédente. En cliquant plusieurs fois sur cette option, vous pouvez revenir jusqu'à 10 tables en arrière. S'il n'y a plus de table précédente à afficher, cette option est désactivée.

Icône	Option	Description
	Table suivante	Si vous êtes revenu en arrière dans la séquence de tables affichées en utilisant l'option <i>Table précédente</i> , cette option est activée pour avancer dans la séquence des tables MRL affichées. Si vous avez atteint la dernière table (la plus récente) dans la séquence, cette option est désactivée.
	Sauter à la rangée	Cliquez sur ce bouton pour sauter à une rangée spécifique dans la table couramment affichée. La fenêtre suivante s'affiche:



Saisissez le numéro de rangée requis et cliquez sur le bouton OK.

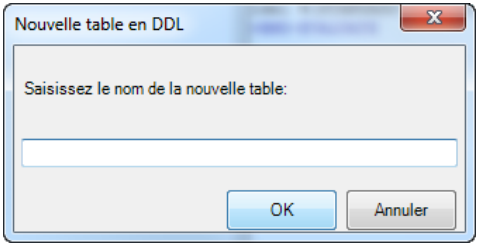
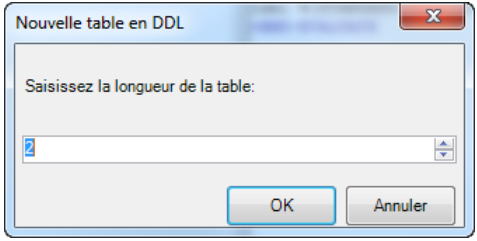
En conséquence, la rangée sélectionnée est surlignée dans le fichier:

```
@TNR RL1012 A F 72
>EVALUATE '#5'
>WHEN 'STD'
lrem - no removal - external source is no copy
>WHEN 'FTP'
>WHEN 'CPY' → ligne 5
lrem - remove external source %[STANDARD-PREFIX]F#2%
ldel %[STANDARD-PREFIX]F#2%
>END-EVALUATE
```

Note: Les rangées commençant par @ seront ignorées.

- Modifiez le code si nécessaire.

Note: Si vous cliquez avec le bouton droit sur le contenu de la table, un menu déroulant s'affichera contenant plusieurs commandes telles que Copier, Coller, Imprimer, ...

Option	Description
Nouveau	Cliquez avec le bouton droit sur le contenu de la table et sélectionnez <i>Nouveau</i> dans le menu. La fenêtre suivante s'affiche:  Saisissez le nom de la nouvelle table. Il doit être composé d'exactly 6 caractères. Ensuite cliquez sur le bouton <i>OK</i>. La fenêtre suivante s'affiche:  Saisissez le nombre de rangées attendues et cliquez sur le bouton <i>OK</i>. La table est créée. Maintenant, vous pouvez saisir la codification nécessaire et sauvegarder la table vers un fichier DDL en utilisant l'option <i>Sauvegarder en DDL</i>.

6. Une fois que vous avez fait les modifications nécessaires, vous pouvez utiliser une des options suivantes: *Sauvegarder en DDL*, *Sauvegarder en MIL* ou *Générer*.

- Sauvegarder en DDL

Cliquez sur l'option *Sauvegarder en DDL* pour sauvegarder la version actuellement affichée d'une table vers le fichier **MSMRL.DDL**.

Cela est nécessaire:

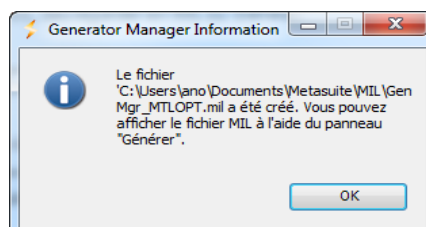
- après avoir modifié un fichier existant
- après avoir créé un nouveau fichier

La nouvelle version de la table est sauvegardée dans le fichier **MSMRL.DLL**.

- Sauvegarder en MIL

Cliquez sur le bouton *Sauvegarder en MIL* pour sauvegarder les modifications d'une table vers un nouveau fichier MIL dans le répertoire standard des fichiers MIL.

La fenêtre suivante s'affiche:



Le format du nom du fichier est comme suit:

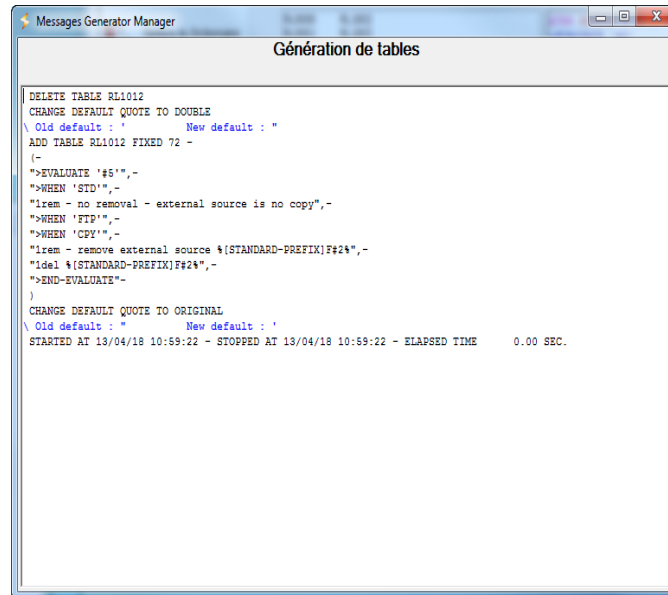
GenMgr_TableName.MIL

Implémentez le Fichier MIL pour rendre les changements effectifs. Voir [Copier des Fichiers MIL existants](#) on page 20.

- Générer

Cliquez sur le bouton *Générer* pour rendre les changements immédiatement effectifs dans le Dictionnaire du Générateur.

La fenêtre suivante s'affiche:



Les autres fonctions

La fenêtre *Autres fonctions* permet d'accéder aux paramètres généraux initiaux et aux dossiers par défaut.

Accédez à l'écran *Autres fonctions* en cliquant sur l'icône appropriée dans la colonne de gauche.

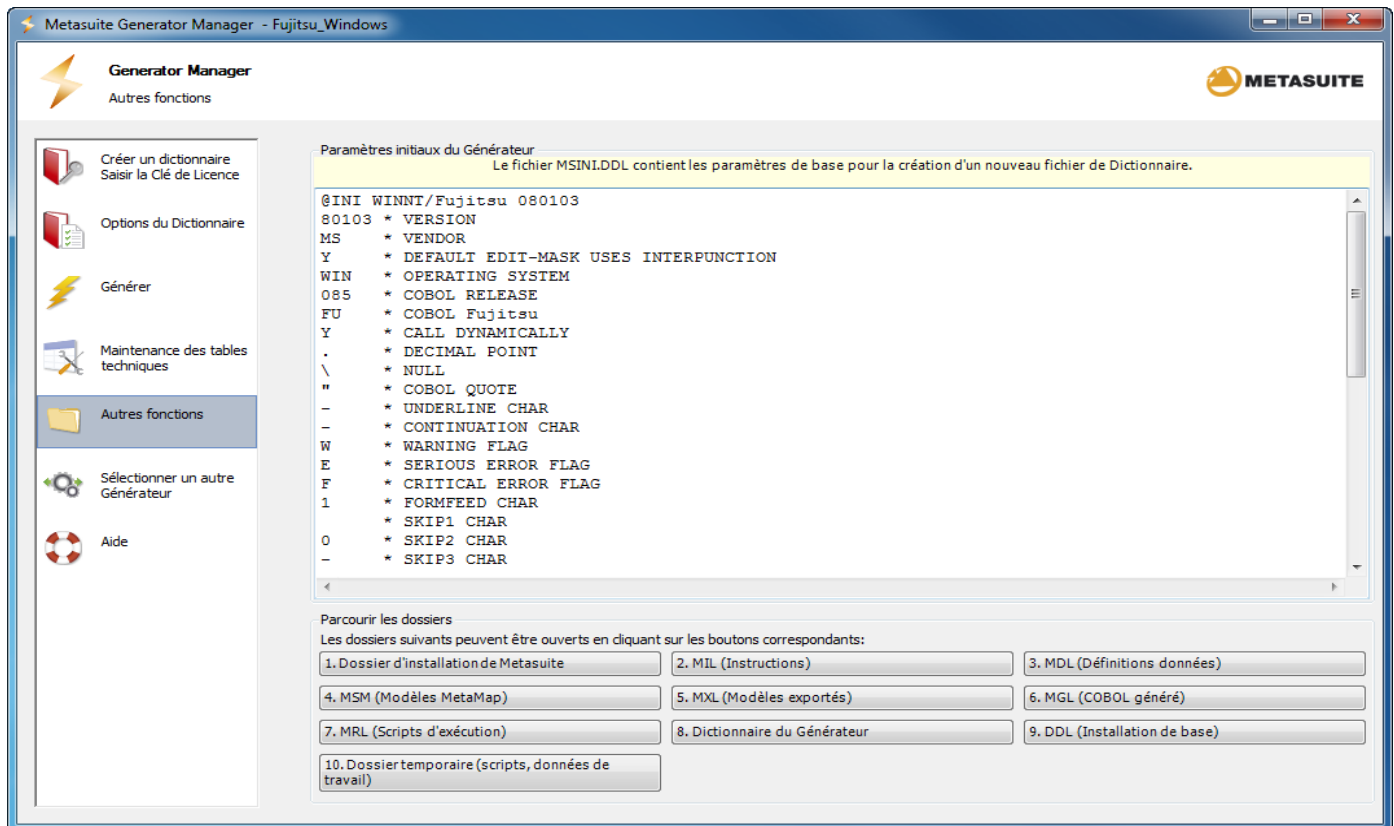
Pour une description détaillée, se référer aux procédures suivantes:

- [Paramètres initiaux du Générateur](#) (page 84)
- [Parcourir les dossiers](#) (page 85)

13.1. Paramètres initiaux du Générateur

Les paramètres initiaux sont utilisés pour la création d'un nouveau Fichier de Dictionnaire.

1. Accédez à l'écran *Autres fonctions* en cliquant sur l'icône appropriée dans la colonne de gauche.
Les paramètres initiaux du Générateur sont affichés en haut à gauche.

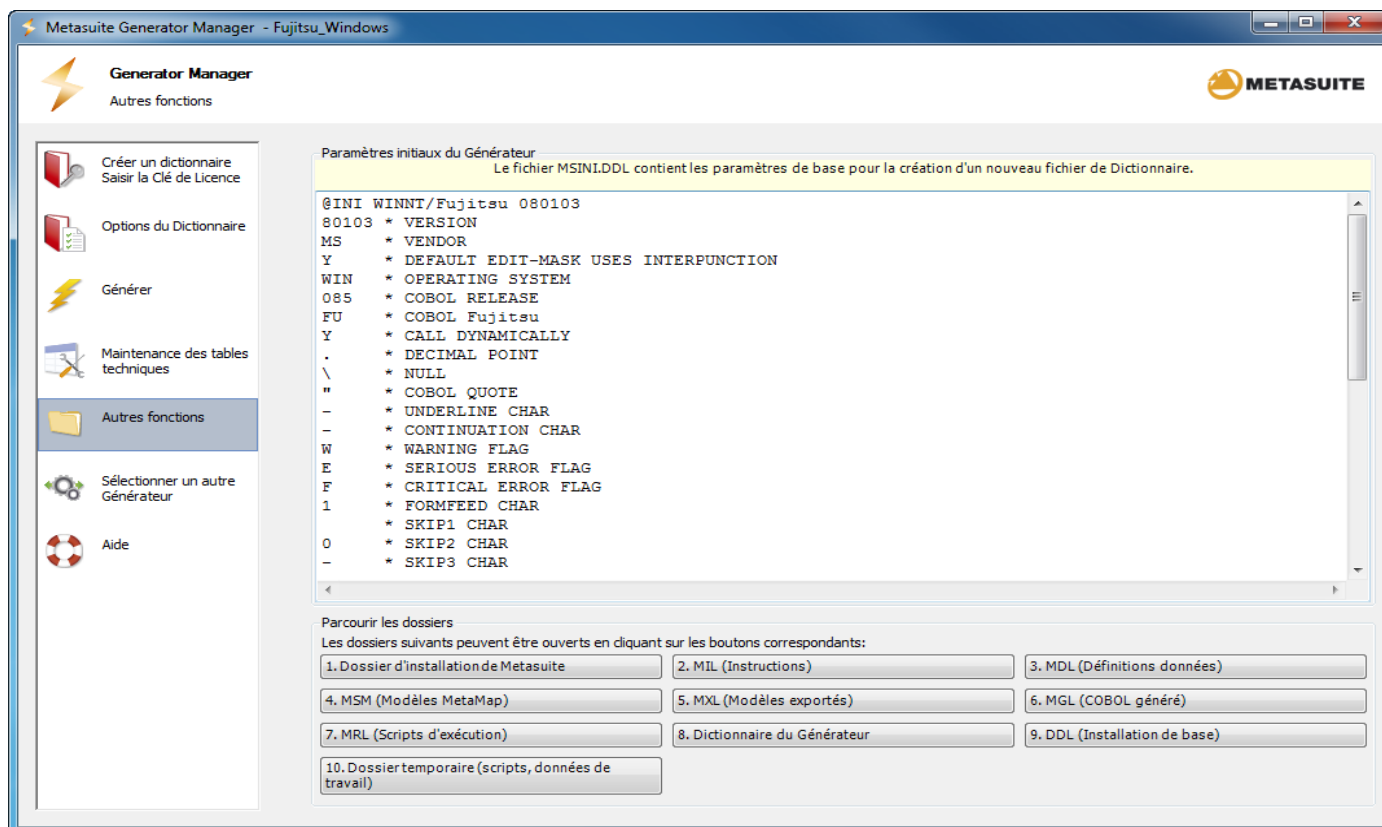


2. Vérifiez les paramètres.

13.2. Parcourir les dossiers

Cette option permet d'afficher rapidement le dossier d'Installation MetaSuite et les dossiers par défaut contenant les fichiers associés à la gestion du Dictionnaire du Générateur pour le Générateur sélectionné. Ces dossiers par défaut sont spécifiés dans INI Manager.

1. Accédez à l'écran *Autres fonctions* en cliquant sur l'icône appropriée dans la colonne de gauche. Ces options permettent d'afficher les dossiers par défaut affichés en bas à gauche.



Vous pouvez afficher le contenu des dossiers suivants:

Dossier	Contenu
Installation MetaSuite	Les fichiers d'installation principaux
Dossier MIL	Les Fichiers MIL contiennent les instructions qui peuvent immédiatement être implémentées dans le Dictionnaire du Générateur. Voir Générer - Implémenter les instructions MIL on page 18.
Dossier MDL	Les Fichiers MDL contiennent les définitions de Métadonnées pour les Sources de données et les Cibles de données. Voir Générer les fichiers MDL on page 37.
Dossier MSM	Les Fichiers MSM contiennent les règles de transformation définies dans un Modèle MetaMap, dans un format codifié. Voir Générer les fichiers MSM on page 47.
Dossier MXL	Les Fichiers MXL contiennent les règles de transformation définies dans un Modèle MetaMap, dans un format lisible. Voir Générer les fichiers MXL on page 49.

Dossier	Contenu
Dossier MGL	Les Fichiers MGL contiennent le Code Source COBOL généré sur la base du Modèle MetaMap. Voir Maintenance des tables techniques - Maintenance des tables MGL on page 58.
Dossier MRL	Les Fichiers MRL contiennent les commandes de script d'exécution pour le Code Source COBOL (Fichier MGL), qui résulte de la génération d'un Modèle MetaMap. Voir Maintenance des tables techniques - Maintenance des tables MRL on page 78.
Dictionnaire du Générateur	Les Fichiers DCT ou les Fichiers de Dictionnaire du Générateur contiennent les règles utilisées pour la génération du Code Source COBOL et des scripts d'exécution COBOL sur la base du Modèle MetaMap. Les Fichiers de Dictionnaire ne peuvent pas être modifiés directement. Pour plus d'informations, se référer aux sections suivantes: • Créer des Fichiers de Dictionnaire (page 6) • Implémenter des instructions MIL (page 18) • Modifier les options MTL (page 12) • Modifier la Clé de licence (page 9) • Modifier les Tables MRL (page 78) • Modifier les Tables MGL (page 58)
Dossier DDL	Il y a trois Fichiers DDL: • MSINI.DDL • MSMRL.DDL • MSMGL.DDL Ces Fichiers sont utilisés pour créer des nouveaux Dictionnaires de Générateur. Les Fichiers DDL ne peuvent pas être modifiés directement. Pour plus d'informations, se référer aux sections suivantes: • Créer un nouveau Fichier de Dictionnaire (page 6) • Modifier les Tables MRL (page 78) • Modifier les Tables MGL (page 58)
Dossier Temporaire	La génération et la compilation ont lieu dans le dossier temporaire.

2. Afficher le contenu du dossier requis.

Si vous cliquez sur un des boutons de dossier, le dossier s'ouvre dans l'Explorateur Windows.

Le tableau suivant affiche les emplacements de ces dossiers.

Dossier	Emplacement par défaut
Installation MetaSuite	<Installation>
MIL (Instructions)	<Installation>\MIL
MDL (Définitions données)	<Installation>\MDL
MSM (Modèles MetaMap)	<Installation>\MSM
Dossier MXL (mappages exportés)	<Installation>\MXL
MGL (COBOL généré)	<Installation>\<GEN>\<MGL
MRL (Scripts d'exécution)	<Installation>\<GEN>\MRL
Dictionnaire du Générateur	<Installation>\<GEN>\DCT
DDL (Installation de base)	<Installation>\<GEN>\DDL
Dossier temporaire (scripts, données de travail)	<Installation>\<GEN>\TMP

Note: Si requis, vous pouvez modifier les emplacements par défaut dans le fichier *MetaSuite.ini* en utilisant *MetaSuite INI Manager*.

Commandes du langage d'installation

Les commandes de Langage d'installation suivantes sont disponibles:

- [ADD TABLE](#) (page 95)
- [CHANGE DEFAULT BUILD](#) (page 89)
- [CHANGE DEFAULT CHARACTER-CCSID](#) (page 94)
- [CHANGE DEFAULT DATE](#) (page 89)
- [CHANGE DEFAULT DECIMAL](#) (page 90)
- [CHANGE DEFAULT DYNAMIC](#) (page 94)
- [CHANGE DEFAULT EXEC](#) (page 91)
- [CHANGE DEFAULT INTERPUNCTION](#) (page 91)
- [CHANGE DEFAULT NULL](#) (page 92)
- [CHANGE DEFAULT NULLABLE](#) (page 92)
- [CHANGE DEFAULT PAGE](#) (page 93)
- [CHANGE DEFAULT QUOTE](#) (page 93)
- [CHANGE DEFAULT SQL DIALECT](#) (page 94)
- [CHANGE DEFAULT SQL-QUOTE](#) (page 94)
- [CHANGE DEFAULT UNICODE-CCSID](#) (page 94)
- [COPY TABLE](#) (page 95)
- [DELETE TABLE](#) (page 96)
- [LIST DEFAULT](#) (page 96)
- [LIST TABLE](#) (page 96)
- [LIST VERSION](#) (page 97)
- [NEW](#) (page 89)
- [REMARKS](#) (page 98)

14.1. NEW

NEW NNNN

La commande NEW crée un nouveau Dictionnaire de Générateur. Cette commande doit être la première commande dans un Fichier MIL si vous démarrez le Générateur pour exécuter le chargement initial d'un nouveau Dictionnaire du Générateur.

Colonnes 1-3

Obligatoire.

Les colonnes 1-3 doivent contenir le mot clé NEW.

nnnn

Obligatoire.

Les colonnes 5-8 contiennent un nombre à quatre chiffres (y compris les zéros en tête, si présents) représentant le nombre de blocs du Dictionnaire du Générateur à initialiser.

14.2. CHANGE DEFAULT BUILD

CHANGE DEFAULT BUILD NUMÉRO-CONSTRUCTION

La commande CHANGE DEFAULT BUILD est utilisée pour modifier le numéro de construction du Dictionnaire après avoir appliqué un "patch".

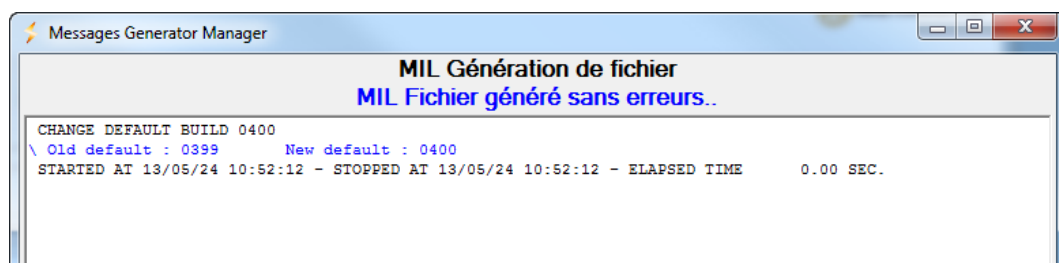
Grâce à cette commande, la génération d'un nouveau Dictionnaire peut être évitée.

Numéro de construction

Obligatoire.

Il s'agit d'un numéro de construction entre 00 et 99.

Exemple



14.3. CHANGE DEFAULT DATE

FORMAT 1:

CHANGE DEFAULT DATE { 'ISO' | 'EUR' | 'JIS' | 'USA' }

La commande CHANGE DEFAULT DATE est utilisée pour définir le format de date à utiliser pour la manipulation de dates dans un environnement RDBMS.

ISO

Toutes les manipulations de date dans un environnement RDBMS sont faites en format ISO (YYYY?MM?DD).

Lors de la sortie d'une date, le séparateur de date est un trait d'union. (2002-10-30)

EUR

Toutes les manipulations de date dans un environnement RDBMS sont faites en format EUR (DD?MM?YYYY).

Lors de la sortie d'une date, le séparateur de date est un point. (30.10.2002)

JIS

Toutes les manipulations de date dans un environnement RDBMS sont faites en format JIS (YYYY?MM?DD).

Lors de la sortie d'une date, le séparateur de date est un trait d'union. (2002-10-30)

USA

Dans l'environnement RDBMS, toutes les manipulations de données se font en format USA (MM?DD?YYYY).

Lors de la sortie d'une date, le séparateur de date est une barre transversale. (10/30/2002)

(siècle, année-bascule)

FORMAT 2:

CHANGE DEFAULT DATE (SIÈCLE , ANNÉE-BASCULE)

Cette commande est utilisée pour spécifier si une année appartient au 20^{ième} ou au 21^{ième} siècle si le format de date n'inclut pas le siècle comme par exemple le format de date YYMMDD.

Pour définir le siècle par défaut pour les formats de date n'utilisant que deux chiffres pour l'année, tels que YYMMDD, spécifiez les deux chiffres identifiant le siècle. Par exemple, 19 spécifie le 20^{ième} siècle et formatera l'année comme 19nn; 20 spécifie le 21^{ième} siècle et formatera l'année comme 20nn. Le défaut installé pour le siècle est 19.

Pour définir le point de bascule pour le siècle par défaut, spécifiez l'option "année-bascule" comme un nombre à deux chiffres qui identifie l'année à laquelle le nouveau siècle doit commencer au cas où les années ne sont spécifiées que par un nombre à deux chiffres (p.ex., yy). Par exemple, si "19" est spécifié comme étant le siècle et le point de bascule a été établi à "55", les années qui sont égales à ou plus élevées que l'année spécifiée sont formatées comme "19nn" et les années inférieures à l'année de fin sont formatées comme "20nn". La commande suivante établit le siècle à "19" et l'année de fin à "80" de sorte que les dates pour lesquelles uniquement l'année (yy) a été spécifiée sont supposées se situer dans les plages "1980 à 1999" et "2000 à 2079".

CHANGE DEFAULT DATE (19,80)

14.4. CHANGE DEFAULT DECIMAL

CHANGE DEFAULT DECIMAL [TO { POINT | COMMA }]

La commande CHANGE DEFAULT DECIMAL est utilisée pour modifier le caractère décimal par défaut d'un point décimal (.) en une virgule décimale (,), et vice versa.

La commande CHANGE DEFAULT DECIMAL TO COMMA est utilisée pour établir le caractère décimal par défaut à une virgule décimale.

La commande `CHANGE DEFAULT DECIMAL TO POINT` est utilisée pour établir le caractère décimal par défaut à un point décimal.

Le système est installé avec le Point décimal (.) comme défaut.

Le caractère décimal a un impact sur le MGL généré, vu qu'il spécifie la représentation du caractère décimal pour les valeurs des Champs Source ainsi que les valeurs des Champs Cible qui représentent un Champ numérique mais qui ont été définis comme un Champ alphanumérique.

14.5. CHANGE DEFAULT EXEC

CHANGE DEFAULT EXEC TO { IMS | RESTARTABLE | NULL }

La commande `CHANGE DEFAULT EXEC TO IMS` est utilisée pour permettre le redémarrage sous IMS/DC, ou pour une base de données RDBMS.

Un synonyme de "IMS" est "REDÉMARRABLE". Ce mot est plus facilement accepté si vous utilisez cette option dans un environnement non IMS.

Vous pouvez adapter les noms des programmes de redémarrage standard en utilisant les options MTL.

L'option MTL "CHECKPOINT-CALL" contient le nom du sous-programme COBOL qui sauvegarde les informations de l'Enregistrement du dernier Enregistrement mis à jour. Dans un environnement non IMS, elle est établie à `MSRSTvvv` (vvv=numéro de version MetaSuite).

Cette valeur peut être réinitialisée en utilisant la commande `CHANGE DEFAULT EXEC TO NULL`. Vous ne devez pas réinitialiser le Dictionnaire technique.

Cette option est prise en charge pour OS390 avec IMS, et pour les autres Systèmes d'exploitation utilisant des bases de données RDBMS.

14.6. CHANGE DEFAULT INTERPUNCTION

CHANGE DEFAULT INTERPUNCTION [TO { OFF | NO | YES }]

La commande `CHANGE DEFAULT INTERPUNCTION` est utilisée pour spécifier le masque d'édition par défaut pour les valeurs numériques (si la propriété `CODE` n'est pas spécifiée).

Si la propriété `CODE` est spécifiée pour un champ numérique, le champ sera présenté avec les zéros en tête et sans inter-punctuation. Il sera considéré comme du code, et pas comme une valeur numérique qui peut être comptée.

Pour plus d'informations concernant la propriété `CODE`, se référer aux sections *Définir les Champs* pour le Fichier de Dictionnaire approprié dans le *Guide de l'Utilisateur de MetaStore Manager*.

Si aucune valeur d'inter-punctuation n'est spécifiée, l'inter-punctuation est modifiée de "YES" en "NO", et de toute autre valeur en "YES".

OFF

Si l'option `INTERPUNCTION` est établie à "OFF", chaque champ numérique sera traité comme du code, signifiant: pas de suppression de zéros et pas d'inter-punctuation.

NO

Si l'option `INTERPUNCTION` est établie à "NO", le masque d'édition par défaut pour les champs numériques (non code) spécifiera la suppression de zéros mais pas d'inter-punctuation.

(Les valeurs numériques de code n'auront pas de suppression de zéros ni d'inter-punctuation.)

YES

Si l'option INTERPUNCTION est établie à "YES", le masque d'édition par défaut pour les champs numériques (non code) spécifiera la suppression de zéros ainsi que l'inter-ponctuation.

(Les valeurs numériques de code n'auront pas de suppression de zéros ni d'inter-ponctuation.)

14.7. CHANGE DEFAULT NULL

CHANGE DEFAULT NULL 'CARACTÈRE-NULL'

La commande CHANGE DEFAULT NULL est utilisée pour spécifier le caractère qui sera utilisé pour indiquer la valeur Null dans un Champs d'un Fichier séquentiel.

Le Caractère Null à utiliser pour le stockage INBOUND et OUTBOUND NULL dans un Champ. Si aucun défaut n'est spécifié explicitement, la valeur par défaut du caractère Null est établie à '\'.

Caractère Null

Obligatoire. Le Caractère Null doit être un seul caractère. Dans le cas de INBOUND NULL, vous devez choisir le Caractère Null avec précaution vu que l'apparition du Caractère NULL à la première position du champ détermine si le champ contient une valeur pour Null.

Les valeurs spéciales suivantes sont permises:

SYS-LOW-VALUE ou LOW-VALUE : remplace hex "00".

SYS-HIGH-VALUE ou HIGH-VALUE : remplace hex "FF".

TAB-CHARACTER : remplace hex "09".

14.8. CHANGE DEFAULT NULLABLE

CHANGE DEFAULT NULLABLE 'Option Nullable'

La commande CHANGE DEFAULT NULLABLE est utilisée pour spécifier l'option NULLABLE par défaut pour les champs dans un fichier séquentiel.

Si l'option NULLABLE est établie à 'N', tous les champs pour lesquels aucun paramètre NULL-INDICATOR spécifique n'est spécifié sera traité comme NOT-NULLABLE (c.-à-d., NULL-INDICATOR 'NOTNULL').

Le caractère Null ne sera pas interprété pour ces champs.

Si l'option NULLABLE est établie à 'I', tous les champs pour lesquels aucun paramètre NULL-INDICATOR spécifique n'est spécifié sera traité comme INBOUND-NULLABLE (c.-à-d., NULL-INDICATOR 'INNULL').

L'option Nullable

Obligatoire.

L'option Nullable peut être établie à 'I' ou 'N'.

'N' est interprété comme NonNull, 'I' est interprété comme InNull.

Vous devez choisir votre option Nullable avec précaution car l'impact de cette option est très important.

La valeur standard pour cette option est 'N'.

14.9. CHANGE DEFAULT PAGE

CHANGE DEFAULT PAGE (N , M)

La commande CHANGE DEFAULT PAGE est utilisée pour définir la largeur et la longueur de page par défaut, qui sont utilisées pour les Rapports Cible si aucun paramètre de page n'est spécifié par le Modèle MetaMap.

N se réfère à la longueur de page par défaut (le défaut est établi à 80 lignes),

M se réfère à la largeur de page par défaut (le défaut est établi à 80 caractères par ligne),

14.10.CHANGE DEFAULT QUOTE

CHANGE DEFAULT QUOTE [TO {SINGLE|DOUBLE|ORIGINAL}]

La commande CHANGE DEFAULT QUOTE est utilisée pour modifier les guillemets par défaut de guillemets simples (') en guillemets doubles ("), et vice versa.

CHANGE DEFAULT QUOTE TO SINGLE modifie les guillemets par défaut en guillemets simples.

CHANGE DEFAULT QUOTE TO DOUBLE modifie les guillemets par défaut en guillemets doubles.

CHANGE DEFAULT QUOTE TO ORIGINAL réinitialisé les guillemets par défaut à sa valeur initiale au moment où le processus de génération a démarré.

La plupart des systèmes sont installés avec les guillemets simples comme défaut.

Cependant, la plupart des tables et des "patches" utilisent les guillemets doubles comme défaut.

Le paramètre par défaut peut être trouvé dans le fichier MSINI.DLL qui est utilisé pour initialiser le Dictionnaire du Générateur MetaSuite. Vous pouvez l'afficher en utilisant la commande

"LIST DEFAULT ALL".

Toutes les constantes non numériques dans vos Modèles MetaMap doivent être délimitées avec les guillemets par défaut.

MetaMap Manager utilisera les guillemets simples comme délimiteur pour les constantes non numériques.

La principale utilisation de la commande CHANGE DEFAULT QUOTE est de modifier temporairement les guillemets par défaut, avant d'ajouter ou de modifier une ou plusieurs définitions de table.

Il est recommandé de ne pas modifier les guillemets sur une base permanente en utilisant cette commande, parce que si vous générez le dictionnaire, le paramètre original par défaut sera réinitialisé au contenu trouvé dans le Fichier MSINI.DDL.

Si vous modifiez les guillemets par défaut avant de saisir la commande ADD TABLE, vous pouvez imbriquer des guillemets simples dans le texte de toute entrée de table à ajouter ou à remplacer. Après avoir saisi toutes les commandes de table, vous devez remettre en place le défaut en codifiant la commande CHANGE DEFAULT QUOTE TO ORIGINAL.

La séquence de commandes suivante illustre l'usage de la commande CHANGE DEFAULT QUOTE lors de la modification de tables de bibliothèque:

CHANGE DEFAULT QUOTE TO DOUBLE

DELETE TABLE SALES-HIST-READ

ADD TABLE SALES-HIST-READ 4 -

(-

"BCALL ""HSTRDR03"" USING F#1-RD0001.",-

"BIF F#1-RD0001 = SPACE EXIT."-

)

CHANGE DEFAULT QUOTE TO ORIGINAL

14.11.CHANGE DEFAULT SQL DIALECT

```
CHANGE DEFAULT SQL DIALECT [TO]
{
DB2_400 | DB2_MVS | DB2_LUW | DB2_2 | DB2_VSE | INFORMIX | INGRES | MYSQL | ODBC | ORACLE | RDB | SESAM
| SQLSERVER | SYBASE | TERADATA
}
```

La commande CHANGE DEFAULT SQL DIALECT est utilisée pour spécifier le dialecte SQL préféré à utiliser lors de la génération de SQL imbriqué et qu'aucun dialecte SQL spécifique n'est choisi par MetaMap Manager.

La séquence de commandes suivante illustre l'usage de la commande CHANGE DEFAULT SQL DIALECT lors de la modification de tables de bibliothèque:

```
CHANGE DEFAULT SQL DIALECT TO ORACLE
```

14.12.CHANGE DEFAULT SQL-QUOTE

La commande CHANGE DEFAULT SQL-QUOTE est utilisée pour spécifier les guillemets SQL préférés à utiliser lors de la génération de SQL imbriqué et si aucun guillemet SQL spécifique n'est choisi par MetaMap Manager.

La séquence de commandes suivante illustre l'usage de la commande CHANGE DEFAULT SQL-QUOTE lors de la modification de tables de bibliothèque:

```
CHANGE DEFAULT SQL-QUOTE [TO {SINGLE|DOUBLE|ORIGINAL} ]
```

CHANGE DEFAULT SQL-QUOTE TO SINGLE modifie les guillemets par défaut pour les commandes SQL en guillemets simples.

CHANGE DEFAULT SQL-QUOTE TO DOUBLE modifie les guillemets par défaut pour les commandes SQL en guillemets doubles.

CHANGE DEFAULT SQL-QUOTE TO ORIGINAL réinitialisé les guillemets par défaut pour les commandes SQL à sa valeur initiale au moment où le processus de génération a démarré.

14.13.CHANGE DEFAULT DYNAMIC

Le programme MetaSuite peut être généré de trois façons:

1. les modules d'exécution sont appelés de manière statique. (Liaison statique)
2. les modules d'exécution sont appelés de manière dynamique. (Liaison dynamique)
3. les modules d'exécution sont appelés en interne: les modules d'exécution font partie du Fichier Source. Tout est compilé et lié en une seule fois.

14.14.CHANGE DEFAULT CHARACTER-CCSID

Le CCSID par défaut pour chaque Champ de caractères à un seul octet.

14.15.CHANGE DEFAULT UNICODE-CCSID

Le CCSID par défaut pour chaque Champ Unicode.

14.16.ADD TABLE

ADD TABLE NOM-TABLE FIXED TAILLE-ENTRÉE ('ENTRÉE-TABLE',...)

Il existe trois types de base pour les tables de bibliothèque: Tables Système, Tables de Code-Contrôle et Tables de code de prototype.

Les tables système sont utilisées pour les mots réservés, les messages d'erreur et les tables de paramètres spécifiques au système. Les Tables de Code-Contrôle pour les Fichiers Source sont définies en conjonction avec l'option CODE-CONTROL des commandes ADD FILE. Les tables de code prototype contiennent du COBOL prototype (référéncé dans les Tables de Code-Contrôle) qui sera inséré dans les programmes COBOL générés, en général avec quelques modifications. La commande ADD TABLE est utilisée pour ajouter une table au Dictionnaire du Générateur. Cette commande peut être utilisée pour ajouter tout type de table, mais, typiquement, elle n'est utilisée que pour définir des Tables de Code-Contrôle et les Tables de Code-Contrôle de prototype associées.

Nom de table

Obligatoire. Le "Nom de table" est le nom de la table étant définie dans la bibliothèque. Le nom doit être unique dans la bibliothèque; peut aller jusqu'à six caractères de longueur; il doit commencer par un caractère alphabétique; et il peut contenir les caractères A-Z, 0-9 et les traits d'union imbriqués.

La taille d'entrée

Obligatoire. L'option "Taille d'entrée" indique la longueur maximale, en nombre de caractères, de toute entrée dans la table. Toute entrée plus petite que la "Taille d'entrée" sera complétée à droite par des espaces. La taille d'entrée maximale ne peut pas dépasser 72.

Entrée de table

Obligatoire. Une "Entrée de table" est une chaîne de caractères qui existe en trois formats. Au moins une Entrée de table doit être définie (pour fournir le code qui sera développé dans le programme généré). Chaque Entrée de table doit être placée entre guillemets et doit être séparée de l'entrée suivante par une virgule. Si une Entrée de table contient des guillemets imbriqués, vous devez utiliser la commande CHANGE DEFAULT QUOTE avant d'exécuter la commande ADD FILE pour modifier les guillemets par défaut en guillemets doubles. Ensuite, placez toute entrée de table entre guillemets doubles et reconvertissez les guillemets vers des guillemets simples en codifiant une autre commande CHANGE DEFAULT QUOTE après la commande ADD TABLE. Chacune des trois formats d'entrée de table possibles est décrite plus en détail ci-dessous.

Les tables système ne sont jamais ajoutées à la bibliothèque sauf si une correction du logiciel de génération vous impose de supprimer une table système existante et de la remplacer par une nouvelle version de la même table. Soyez extrêmement prudent lors du codage des entrées de tables système. Les entrées codifiées de manière incorrecte dans une table système peuvent rendre votre bibliothèque inutilisable.

14.17.COPY TABLE

COPY TABLE NOM-TABLE

La commande COPY TABLE est utilisée pour copier un ou plusieurs tables de dictionnaire du Générateur MetaSuite vers un fichier de commandes de sortie. Le fichier MSCOPY.MIL contient les commandes copiées. Ce fichier se trouve dans votre variable TEMP.

Nom de table

Obligatoire. Le "Nom de table" est le nom de la Table de Dictionnaire du Générateur à copier dans le fichier de commandes de sortie. Vous pouvez demander des tables systèmes ou de tables codifiées par l'Utilisateur. Par exemple, pour copier la définition de la table HSTCTL vers le fichier de commandes de sortie, vous pouvez utiliser la commande suivante:

```
COPY TABLE HSTCTL
```

14.18.DELETE TABLE

DELETE TABLE NOM-TABLE

La commande DELETE TABLE est utilisée pour supprimer une table du Dictionnaire du Générateur. Soyez prudent à ne pas supprimer de tables système de la bibliothèque, car cela pourrait rendre le système inutilisable. Pour vous assurer de pouvoir récupérer d'une telle erreur, vous pouvez utiliser la commande COPY TABLE (pour copier la définition de la table à supprimer vers un fichier de sauvegarde) avant d'exécuter la commande DELETE TABLE. S'il devient nécessaire, vous pouvez restaurer la définition de table à partir du fichier de sauvegarde.

Nom de table

Obligatoire. Le "Nom de table" est la table du Dictionnaire du Générateur qui doit être supprimée. Par exemple, pour supprimer la table nommée HSTCLS à partir de la bibliothèque, vous pouvez utiliser la commande suivante:

```
DELETE TABLE HSTCLS
```

14.19.LIST DEFAULT

LIST DEFAULT

La commande LIST DEFAULT est utilisée pour afficher la liste de toutes les valeurs par défaut qui s'appliquent au Dictionnaire du Générateur.

Exemple

```
LIST DEFAULT \Character Set :
\ Decimal Point .
\ Quote '
\ Null \
\Nullable : NOTNULL
\Maximum line length : 00080
\Maximum page length : 00080
\Dialecte SQL : ODBC
\COBOL Calls : Dynamic
>Edit Mask Interpunction : Oui
\Executive : Null
\Default Date Format : ISO (19,00)
```

14.20.LIST TABLE

LIST TABLE NOM-TABLE

La commande LIST TABLE est utilisée pour afficher une table au Dictionnaire du Générateur. C'est commande est particulièrement utile pour produire une copie papier de la liste de chaque table de Dictionnaire du Générateur avant de la modifier ou de la supprimer, si jamais il s'avère nécessaire de restaurer la table à son état d'origine.

Nom de table

Obligatoire. Le "Nom de table" identifie le nom de la table de Dictionnaire du Générateur que vous voulez afficher.

Exemple

```
LIST TABLE AV004
  TABLE NAME REPORT
TABLE LENGTH BLOCK NO OFFSET
*****
AV0045 0072 0005 045
  ENTRIES:
001 >EVALUATE '[SYS-COBOL]/[SYS-OPERATING-SYSTEM]'
002 >WHEN 'VS/VSE'
003 >WHEN 'VS/MVS'
004 B ASSIGN TO DA-I-PPTF#1
005 >WHEN 'VA/UX '
006 >WHEN 'VA/NT '
007 B ASSIGN TO PPTF#1
008 >WHEN '04/OS4'
009 B ASSIGN TO DISK-PPTF#1
010 >WHEN 'C2/BS2'
011 B ASSIGN TO "PPTF#1"
012 >WHEN 'DG/VMS'
013 B ASSIGN TO PPT$F#1
014 >WHEN OTHER
015 B ASSIGN TO F#1-FILENAME
016 >END-EVALUATE
```

14.21.LIST VERSION

LIST VERSION

La commande LIST VERSION est utilisée pour afficher la version et le numéro de construction du Générateur courant, et pour afficher la version et le numéro de construction du Dictionnaire du Générateur.

Le résultat de cette commande contient également les exigences du Générateur et du Dictionnaire du Générateur: une construction spécifique d'un Générateur correspond à un numéro de construction spécifique du Dictionnaire, et vice versa.

Exemple

```
LIST TABLE AV0045
\ TABLE NAME REPORT
\ TABLE LENGTH BLOCK NO OFFSET
\ *****
\ AV0045 0048 0005 0603
\ ENTRIES:
\ 001 >EVALUATE '[SYS-COBOL]/[SYS-OPERATING-SYSTEM]'
\ 002 >WHEN 'VS/VSE'
```

```

\ 003 >WHEN 'VS/MVS'
\ 004 >WHEN 'SZ/ZOS'
\ 005 >WHEN 'SZ/WIN'
\ 006 >WHEN 'SZ/UX'
\ 007 B ASSIGN TO DA-I-[STANDARD-PREFIX-NO-DD]F#1
\ 008 >WHEN 'VA/UX '
\ 009 >WHEN 'VA/WIN'
\ 010 B ASSIGN TO [STANDARD-PREFIX-NO-DD]F#1
\ 011 >WHEN '04/OS4'
\ 012 B ASSIGN TO DISK-[STANDARD-PREFIX-NO-DD]F#1
\ 013 >WHEN 'C2/BS2'
\ 014 B ASSIGN TO "[STANDARD-PREFIX-NO-DD]F#1"
\ 015 >WHEN 'DG/VMS'
\ 016 >WHEN 'AC/VMS'
\ 017 B ASSIGN TO [STANDARD-PREFIX-NO-DD]$F#1
\ 018 >WHEN 'MF/UX '
\ 019 >WHEN 'MF/WIN'
\ 020 B ASSIGN TO DYNAMIC F#1-FILENAME
\ 021 >WHEN OTHER
\ 022 B ASSIGN TO F#1-FILENAME
\ 023 >END-EVALUATE
\

```

STARTED AT 13/07/30 16:48:28 - STOPPED AT 13/07/30 16:48:28 - ELAPSED TIME 0.02 SEC.

14.22.REMARKS

REMARKS TEXTE

La commande REMARKS permet d'insérer des lignes de commentaire. Un usage libre de commentaires est conseillé pour pouvoir documenter le fichier de commandes. Le "Texte" consiste en tout segment de texte descriptif. Par exemple, la commande REMARKS suivante illustre clairement ce qui se passe dans la Procédure Fichier suivante:

REMARKS Choisissez une Source RDBMS

Si vous codifiez plusieurs lignes de commentaire, n'oubliez pas d'inclure le caractère de continuation à la fin de chaque ligne de texte à continuer:

REMARKS Choisissez une -

Source RDBMS

Langage de modèle MetaSuite

La bibliothèque du Générateur est une collection de Tables de code COBOL requises pour traduire les Modèles MetaMap (MSM) en une application COBOL pour le Système d'exploitation demandé.

À part les commandes COBOL, ces tables peuvent également contenir des lignes de logique écrites en *Langage de modèle MetaSuite (MTL)*. Ces lignes contiennent le caractère > dans la colonne 1, suivi de la commande MTL actuelle. Le Générateur interprète cette logique MTL avant d'écrire des lignes dans les scripts MGL et MRL générés.

En utilisant MTL, vous pouvez définir des variables, créer des structures conditionnelles et construire des boucles. Les variables de table de code peuvent être incluses dans les instructions MTL et les variables MTL peuvent être incluses dans la codification COBOL des tables de code.

Beaucoup de tables contiennent des variables ou des chaînes de caractères qui sont remplacées par le Générateur pendant le processus de génération. Il existe deux types de variable:

- Variables MTL (placées entre crochets droits).
- Variables du Générateur (composées du signe # suivi d'un seul caractère). Leur valeur n'est pas définie par le code MTL, mais fournie par le Générateur. Cette valeur varie de table en table. La valeur de remplacement peut être de zéro à 32 caractères.

Pour une description détaillée, se référer aux procédures suivantes:

- [Variables, Expressions et Fonctions MTL](#) (page 99)
- [Variables du Générateur](#) (page 104)
- [Commandes MTL](#) (page 105)
- [Variables système MTL](#) (page 121)
- [Variables MTL spécifiques au Fichier Source](#) (page 123)
- [Variables MTL liées à l'Enregistrement Source](#) (page 124)
- [Variables MTL liées au Fichier Cible](#) (page 125)
- [Variables liées à l'Enregistrement Cible](#) (page 126)
- [Variables MTL liées au Champ](#) (page 127)

15.1. Variables, Expressions et Fonctions MTL

Les variables sont des morceaux d'espace mémoire qui peuvent être accédés en utilisant le nom de la variable. L'organisation du contenu variable est définie par le type de variable.

Les Expressions MTL sont des combinaisons de variables, de numéros et d'opérateurs. Elles sont utilisées pour définir ou calculer le contenu d'une variable MTL et elles peuvent être utilisées dans les comparaisons.

Variables MTL

Le nombre de variables MTL utilisées pour générer un seul programme est limité à 1000.

Il est interdit de combiner la syntaxe d'une variable du Générateur et la syntaxe d'une variable MTL. Par exemple: `#[A] OR [[B]]` n'est pas permis.

Le tableau suivant affiche les caractéristiques d'une variable MTL.

Variable	Description
Nom	Le nom doit respecter les règles suivantes: - longueur maximale: 32 caractères - le premier caractère doit être alphabétique - caractères défendus: Espace = # [et] - caractères à éviter: + - / et * - le nom est insensible à la casse.
Type	Une variable MTL peut être: - numérique: les nombres de -99999999 à +99999999 - alphanumérique: la longueur peut être entre 0 et 58 octets.
Contenu	Le contenu d'une variable MTL peut être récupéré à partir de toute entrée de table de code en mettant le nom de la variable entre crochets droits ("[" et "]"). Les variables numériques seront raccourcies par le processus de génération: les espaces de fin seront retirés. les variables alphanumériques ne seront pas raccourcies: les espaces de fin ne seront pas retirés. Le contenu des variables du Générateur et des variables MTL sera remplacé par leurs valeurs avant la vérification de la syntaxe MTL!
Portée	La portée d'une variable n'est pas limitée à une table. La séquence d'interprétation est égale aux lignes apparaissant dans le résultat de sortie MGL et MRL. Les variables MTL définies par une table sont conservées en mémoire de sorte que toutes les tables peuvent les invoquer à un stade ultérieur. (la portée = globale) Cela signifie qu'une variable créée à la ligne 100 du code MGL sera toujours disponible si utilisée à la ligne 2000 du code MGL. En plus, elle sera également disponible au moment de la génération du MRL.

Les variables MTL temporaires

Les variables MTL dont le nom commence par "@" sont des variables temporaires.

La portée d'une variable temporaire est limitée à la table pour laquelle elle est créée, ou à un niveau de CALL plus bas. Pour utiliser la variable MTL temporaire d'un niveau plus élevé, le caractère d'en-tête "@" doit être remplacé par "?".

Exemple:

```
@TNR MTLTST A F 70
>*****
>* TABLE : MTLTST *
>* PURPOSE : TESTING MTL *
>* DRIVER : PPTGGS *
>* CHANGED : FIB 30/11/2011 *
>*****
>* THIS TABLE HAS NO OTHER PURPOSE THAN FOR TESTING MTL
>* INSTRUCTIONS.
>* IT IS THE FIRST TABLE OF THE GENERATION.
>*****
>CALL TABLE1

@TNR TABLE1 A F 70
>SET @NUMBER = 1000
>SET @TEXT = 'THE QUICK BROWN FOX JUMPS...'
```

```

* @NUMBER = [@NUMBER]
* @TEXT = [@TEXT]
>CALL TABLE2
* @NUMBER = [@NUMBER]
* @TEXT = [@TEXT]

@TNR TABLE2 A F 70
*** ENTERING TABLE2 ***
>SET @NUMBER = [?NUMBER]+1
>SET @TEXT = '...OVER THE LAZY DOG'
* @NUMBER = [@NUMBER]
* ?NUMBER = [?NUMBER]
* @TEXT = [@TEXT]
* ?TEXT = [?TEXT]
*** LEAVING TABLE2 ***

RESULT:
* @NUMBER = 1000
* @TEXT = THE QUICK BROWN FOX JUMPS...
*** ENTERING TABLE2 ***
* @NUMBER = 1001
* ?NUMBER = 1000
* @TEXT = ...OVER THE LAZY DOG
* ?TEXT = THE QUICK BROWN FOX JUMPS...
*** LEAVING TABLE2 ***
* @NUMBER = 1000
* @TEXT = THE QUICK BROWN FOX JUMPS...
    
```

Expressions MTL

Le tableau suivant affiche les types d'expression MTL.

Type d'expression	Description
Numérique	Les expressions numériques sont des combinaisons de caractères numériques et d'opérateurs (+, -, * et /). Le générateur va effectuer un calcul sur cette expression, et le résultat de ce processus sera pris en compte. Les variables numériques peuvent être utilisées si elles sont mises entre crochets droits. Exemple: 4+5*6-7*[NUM]/8
Alphanumérique	Les expressions alphanumériques commencent par un guillemet simple ou des guillemets doubles, et se terminent par le même type de guillemet. Si l'expression (la chaîne de caractères) alphanumérique contient un des guillemets enfermant, ce guillemet doit être doublé. Exemple: "ALPHA IS [ALPHA]"
Noms de variable	Si l'expression n'est ni une expression numérique, ni une chaîne de caractères et ni le mot "FUNCTION", le Générateur suppose qu'il s'agit d'un nom de variable. Exemple: AZERTY(1)
Fonction	Si l'expression commence par le mot "FUNCTION" ou "F:", une fonction MTL sera exécutée, et le résultat de cette fonction sera prise comme valeur. Exemple: FUNCTION MID("BRUSSELS BY NIGHT",10,2)

Fonctions MTL

La syntaxe de la commande **FONCTION** est très stricte:

- Immédiatement après le mot "FUNCTION" il peut y avoir un seul espace, suivi du nom de la fonction.
- Le mot "FUNCTION3" peut être remplacé par "F:"
- Le nom de la fonction est suivi des paramètres placés entre parenthèses.
- Les paramètres entre les parenthèses sont séparés par un caractère de délimitation, qui peut être une virgule ou un espace.

Le tableau suivant liste les fonctions MTL.

Fonction	Description
LENGTH	La fonction LENGTH renvoie la longueur d'une chaîne de caractères. Syntaxe: FUNCTION LENGTH("Chaîne de caractères") L'utilisation des guillemets est obligatoire! Exemple: >SET X="TEST" >SET L=FUNCTION LENGTH("[X]") Résultat: L = 4
MIDDLE	La fonction MIDDLE renvoie une partie d'une chaîne de caractères. Syntaxe: FUNCTION MIDDLE("Chaîne de caractères",Position de départ[,Longueur]) Où: • Chaîne de caractères = La chaîne de caractères principale à partir de laquelle vous voulez obtenir la sous-chaîne de caractères • Position de départ = la position de départ de la partie. Ce paramètre doit être un nombre entre un et la longueur de la chaîne de caractères principale. • Longueur = la longueur de la sous-chaîne de caractères. La longueur doit être zéro ou un nombre positif. Si le paramètre de longueur est omis, la longueur sera la longueur restante de la chaîne de caractères principale, commençant à la position de départ. L'utilisation des guillemets est obligatoire dans le premier paramètre! Exemple: >SET X="TEST123" >SET Y=FUNCTION MIDDLE("[X]",3,2) >SET Z=FUNCTION MIDDLE("[X]",5) Résultat: • Y = ST • Z = 123
LEFT	La fonction LEFT renvoie la partie gauche de la chaîne de caractères. Syntaxe: FUNCTION LEFT("Chaîne de caractères",Longueur) Où: • Chaîne de caractères = La chaîne de caractères principale à partir de laquelle vous voulez obtenir la sous-chaîne de caractères • Longueur = la longueur de la sous-chaîne de caractères. La longueur doit être zéro ou un nombre positif. L'utilisation des guillemets est obligatoire dans le premier paramètre! Exemple: >SET Y=FUNCTION LEFT("TEST123",3) Résultat: Y = TES

Fonction	Description
DROITE	La fonction RIGHT renvoie la partie droite de la chaîne de caractères. Syntaxe: FUNCTION RIGHT("Chaîne de caractères",Longueur) Où: • Chaîne de caractères = La chaîne de caractères principale à partir de laquelle vous voulez obtenir la sous-chaîne de caractères • Longueur = la longueur de la sous-chaîne de caractères. La longueur doit être zéro ou un nombre positif. L'utilisation des guillemets est obligatoire dans le premier paramètre! Exemple: >SET Y=FUNCTION RIGHT("TEST123",5) Résultat: Y = ST123
INSTRING	La fonction INSTRING recherche une sous-chaîne de caractères dans une chaîne de caractères. Syntaxe: FUNCTION INSTRING("Chaîne de caractères1","Chaîne de caractères2",[Position de départ]) Où: • Chaîne de caractères1 = La chaîne de caractères principale à partir de laquelle vous voulez obtenir la sous-chaîne de caractères • Chaîne de caractères2 = La chaîne de caractères, qui peut faire partie de la chaîne de caractères principale • Position de départ = la position de départ de l'opération de recherche. Cette position doit être un nombre positif. Si omise, la position de départ est égale à 1. L'utilisation des guillemets est obligatoire dans le premier et le deuxième paramètre! Exemple: >SET Y=FUNCTION INSTRING("BRUSSELS","SEL",3) >SET Z=FUNCTION INSTRING("BRUSSELS","XYZ") Résultats: • Y = 5 (la chaîne de caractères recherchés commence à la position 5). • Z = 0 (chaîne de caractères recherchés non retrouvée)
UPPER	La fonction UPPER convertit tous les caractères alphabétiques en une chaîne de caractères en majuscules. Syntaxe: FUNCTION UPPER("Chaîne de caractères") L'utilisation des guillemets est obligatoire! Exemple: >SET Y=FUNCTION UPPER("Brussels") Résultat: Y = BRUSSELS
LOWER	La fonction LOWER convertit tous les caractères alphabétiques en une chaîne de caractères en minuscules. Syntaxe: FUNCTION LOWER("Chaîne de caractères") L'utilisation des guillemets est obligatoire! Exemple: >SET Y=FUNCTION LOWER("Brussels") L'utilisation des guillemets est obligatoire! Résultat: Y = brussels

Fonction	Description
IS-SET	La fonction IS-SET est utilisée pour tester si une variable a été déclarée ou non par l'instruction SET. Syntaxe: FUNCTION IS-SET(Nom de la variable) L'utilisation des crochets n'est pas obligatoire! Exemple 1: >SET Y=FUNCTION IS-SET(SYS-QUOTE) Résultat: Y = "TRUE" Exemple 2: >SET Z=FUNCTION IS-SET(UNKNOWN-VARIABLE) Z = "FALSE".
VAR-TYPE	La fonction VAR-TYPE est utilisée pour vérifier si une variable MTL est considérée comme numérique, alphanumérique ou non-définie. Le résultat peut être "ALPHA", "NUM" ou "NULL". Syntaxe: FUNCTION VAR-TYPE(Nom de la variable) L'utilisation des crochets n'est pas obligatoire! Exemple 1: >SET Y=FUNCTION VAR-TYPE(SYS-QUOTE) Résultat: Y = "ALPHA" Exemple 2: >SET Z=FUNCTION VAR-TYPE(UNKNOWN-VARIABLE) Z = "NULL".
VAR-LENGTH	La fonction VAR-LENGTH est utilisée pour déterminer la taille d'une variable MTL. Syntaxe: FUNCTION VAR-LENGTH(Nom de la variable)
QUOTE	La fonction QUOTE est utilisée pour doubler les guillemets dans une variable MTL. Cette fonction peut être utilisée pour imbriquer une valeur dans une chaîne de caractères. Syntaxe: FUNCTION QUOTE(Nom de la variable) Exemple: <pre> > SET @DECL-LEN = F:LEN(@DECL) > SET @DECL-QUOT = F:QUOTE (@DECL) B10 FILLER PIC X([@DECL-LEN]) VALUE "[@DECL-QUOT]" </pre>

15.2. Variables du Générateur

Les Variables du Générateur ne sont pas définies par la codification MTL. Elles sont fournies par le Générateur. Leur valeur varie de table en table. La valeur de remplacement peut être de zéro à 32 caractères.

Les Variables du Générateur sont fournies par le Générateur. Leurs noms commencent par le caractère #, suivi d'un des caractères suivants "123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ.²³⁰".

Les Variables du Générateur ne sont pas seulement utilisées dans le code COBOL, mais également dans les expressions MTL. La longueur de ces variables est limitée à 54 caractères.

La variable du Générateur "##" est utilisée comme indicateur du niveau de Champ.

Valeur	Description
#1 to #9	Variable du Générateur numéro 1 à 9
#A to #U	Variable du Générateur numéro 10 à 30
##	Le caractère #
#f	Le Numéro de Séquence. Le numéro indiquant combien de fois cette variable a été appelée.
#μ	Le numéro de ligne COBOL courant dans le code MGL (COBOL) ou le code MRL (langage de script).
#\$	Indicateur du niveau du Champ "___02_" pour une codification "redémarrable"/IMS, et "_01_" pour une autre codification.
©	La version de MetaSuite (code à 3 chiffres).

Le contenu des variables est toujours raccourci par le Générateur: les espaces de fin sont retirés.

Note: Le contenu des variables du Générateur et des variables MTL sera remplacé par leurs valeurs avant la vérification de la syntaxe MTL!

15.3. Commandes MTL

Le tableau suivant liste les commandes MTL disponibles. Pour une description détaillée, se référer aux sections spécifiques:

Commande MTL	Description abrégée
Boucles FOR-NEXT (page 107)	Les boucles <i>FOR-NEXT</i> sont utilisées pour exécuter des boucles structurées. Elles aident le programmeur de table à programmer les boucles plus facilement.
CALL table (page 108)	La commande <i>CALL table</i> permet d'insérer de la logique de table à partir d'une table dans une autre table. Ceci est intéressant si le même morceau de logique revient dans plusieurs situations. L'instruction <i>CALL</i> est utilisée pour simplifier les tables compliquées.
DUMP (page 109)	L'instruction <i>DUMP</i> est utilisée pour déboguer la codification générée.
EXIT Table (page 109)	La commande <i>EXIT</i> permet de quitter une table: • pour retourner à table appelante, si le niveau d'imbrication n'est pas égal à zéro. Pour l'instruction <i>EXIT</i>, les mêmes règles s'appliquent que pour l'instruction <i>GOTO</i>. • Pour continuer avec la table suivante (si le niveau d'imbrication est égal à zéro).
FREEZE et UNFREEZE (page 109)	La commande <i>FREEZE</i> arrête l'interpréteur de la table de code (dont fait partie l'interpréteur MTL) pour exécuter certaines actions standard. <i>UNFREEZE</i> redémarre le traitement des lignes de code normal.

Commande MTL	Description abrégée
GOTO (page 110)	La commande de saut <i>GOTO</i> permet de sauter ou de répéter certaines commandes MTL et certaine codification COBOL dans le code MGL généré.
IF imbriqué (page 111)	Il est possible d'imbriquer des blocs IF et EVALUATE multiples.
IMPLEMENT (page 112)	La commande <i>IMPLEMENT</i> est utiliser pour appeler une section à un stade ultérieur.
REMARK (page 112)	La commande <i>REMARK</i> a un caractère purement informatif. Elle offre à l'Utilisateur la possibilité d'ajouter des commentaires dans la codification MTL.
SET (page 112)	La commande <i>SET</i> crée ou modifie une variable MTL.
SKIP (page 113)	La commande de saut <i>SKIP</i> permet de sauter ou de répéter certaines commandes MTL et certaine codification COBOL dans le code MGL généré.
Structures EVALUATE (page 114)	Les structures <i>EVALUATE</i> sont utilisées si l'action à effectuer dépend de plusieurs valeurs de variable. L'instruction <i>EVALUATE</i> évalue une expression et exécute les instructions correspondant au résultat de l'évaluation.
Structures IF (page 115)	Les structures <i>IF</i> sont nécessaires pour exécuter des actions à condition.
TRACE (page 116)	La commande <i>TRACE</i> est utilisée pour tracer ou déboguer la codification MTL. Elle fournit plus d'informations concernant la logique appliquée par l'interpréteur MTL.
Traitement des messages d'erreur (page 117)	Dans le Langage de modèle MetaSuite vous pouvez programmer vos propres messages d'erreur et arrêter le processus de génération, si nécessaire.
UNSET (page 120)	La commande <i>UNSET</i> retire une variable MTL.

Conventions de notation

Cette section décrit la métasyntaxe utilisée pour décrire les différentes commandes MTL. Chaque commande est composée de mots clé, de mots clé optionnels et de règles de codification.

La méta-syntaxe ci-dessous illustre les conventions de notation utilisées pour décrire une commande MTL.

COMMAND Nom du Champ

[OPTION-ONE (valeur-utilisateur,...)]

[OPTION-TWO {A | B | C [c-modifier]}]

[OPTION-THREE][°]

[OPTION-FOUR]¹

[RULE Texte]

Description:

- Mots clé:

Les mots clé sont considérés comme des mots réservés et ne peuvent pas être utilisés comme le nom d'un Champ de Travail ou le nom de MetaStore. Dans notre exemple, *RULE* est le mot clé.

- Informations variables fournies par l'Utilisateur

Les informations variables fournies par l'Utilisateur sont saisies en minuscules. Dans notre exemple, "Nom du Champ", "Valeur d'utilisateur" et "texte" sont des informations fournies par l'Utilisateur.

- Crochets droits ([]):

Les crochets droits indiquent des composants optionnels de la commande, par exemple les options que vous pouvez inclure ou pas, en fonction des nécessités du traitement. N'incluez pas les crochets dans la codification de votre programme. Sauf quelques exceptions, les options peuvent être codifiées dans n'importe quel ordre. Si utilisée, l'option **RULE**, qui est une option pour chaque commande, doit se trouver en dernier.

Si le crochet droit fermant est suivi d'un 0 en chiffre supérieur, cette option peut être répétée plusieurs fois.

Si le crochet droit fermant est suivi du chiffre 1 en chiffre supérieur, cette option peut être répétée plusieurs fois, et elle doit être utilisée au moins une fois.

- **Parenthèses:**

Dans la codification du programme, les parenthèses doivent être incluses exactement aux endroits indiqués dans la syntaxe. Les parenthèses sont utilisées dans la codification du programme pour entourer des listes d'éléments, comme l'illustre l'exemple précédent.

Une virgule ou des pointillés (trois points) après un élément indiquent qu'une liste d'un ou de plusieurs éléments est attendue, comme l'illustre l'exemple ci-dessus pour **OPTION-ONE**.

Si des éléments multiples sont codifiés, chaque élément doit être séparé du suivant par une virgule.

- **Accolades ({ }):**

Les accolades entourent des choix alternatifs, dont un doit être sélectionné. À l'intérieur des accolades, des barres verticales séparent les choix alternatifs.

Dans notre exemple, si vous choisissez de codifier **OPTION-TWO**, vous devez inclure le mot clé "A", "B" ou "C". N'incluez pas les crochets dans la codification de votre programme.

Boucles FOR-NEXT

Les boucles *FOR-NEXT* sont utilisées pour exécuter des boucles structurées. Elles aident le programmeur de table à programmer les boucles de manière plus facile.

Syntaxe:

```
> FOR variable-1 = valeur-1 TO valeur-2 [ STEP valeur-3 ]
```

```
[... instruction ...]1
```

```
> NEXT [ Variable ]
```

Où:

- **Valeur:**

Toutes les valeurs dans l'instruction **FOR** doivent être numériques.

- **Instruction:**

L'instruction peut être du code COBOL ou une instruction MTL.

- **FOR:**

La première partie est le nom de la variable (**variable-1**) qui variera à chaque fois que la boucle **FOR** est répétée.

Le nom de la *Variable-1* est suivi du signe "=", qui sert de délimiteur.

La deuxième partie, **valeur-1**, doit être numérique. Il s'agit de la première valeur attribuée à la **variable FOR**.

valeur-1 est suivie du mot **TO** qui doit être utilisé comme Délimiteur.

valeur-2 doit également être numérique. Il s'agit de la valeur la plus élevée qui peut être attribuée à la **variable FOR**. Si la valeur de la **variable-1** ne se trouve pas dans la plage de **valeur-1** à **valeur-2**, la boucle **FOR** se termine.

- **STEP:**

La valeur **STEP** est la valeur par laquelle la variable sera modifiée chaque fois que la boucle **FOR-NEXT** est effectuée. La valeur par défaut de **STEP** est 1. Vous pouvez modifier cette valeur par défaut en spécifiant une autre valeur après le mot **STEP** (valeur- 3). La valeur de **STEP** peut être négative.

- **NEXT:**

L'instruction **NEXT** indique la fin de la boucle **FOR**. Tout texte après la commande **NEXT** est purement informatif.

Note: L'imbrication de multiples boucles **FOR** est possible. Jusqu'à 16 niveaux d'imbrication sont autorisés. Il doit y avoir au moins une boucle! Si non, un message d'erreur sera affiché.

Exemple:

```
>FOR I = 1 TO 4
>SET PRODUCT=0
>FOR J = 1 TO 5
>SET PRODUCT=(PRODUCT)+[I]
>SET TABLE([I].[J])=PRODUCT
>IF PRODUCT >= 10
* [I]*[J] IS AT LEAST 10
>END-IF
>NEXT J
>NEXT I
```

Résultats:

```
* 2*5 IS AT LEAST 10
* 3*4 IS AT LEAST 10
* 3*5 IS AT LEAST 10
* 4*3 IS AT LEAST 10
* 4*4 IS AT LEAST 10
* 4*5 IS AT LEAST 10
```

CALL table

La commande **CALL table** permet d'insérer de la logique de table à partir d'une table dans une autre table. Ceci est intéressant si le même morceau de logique revient dans plusieurs situations. L'instruction **CALL** est utilisée pour simplifier les tables compliquées.

Syntaxe:

> **CALL** Nom de la table [Label]

Où:

- **Nom de la table:**

Le nom de la table qui sera appelée. Cette table sera nommée la "table appelée" tandis que la table contenant l'instruction **CALL** sera nommée la "table appelante".

- **Label**

Si l'Utilisateur veut ignorer certaines lignes dans la table appelée, il peut spécifier un label dans la table appelée. Les lignes avant ce label seront ignorées.

Les variables **MTL** et les variables spécifiques à la table (#1, #A, etc...) sont disponibles dans la table appelée. Toutes les variables ont un caractère soi-disant "global". Les variables "locales" ne sont pas prises en charge par **MTL**.

Vous pouvez effectuer une instruction **CALL** dans une table déjà appelée. Les imbrications de **CALL** peuvent aller jusqu'à 16 niveaux. Le niveau d'imbrication est le nombre d'appels imbriqués exécutés. Le niveau d'imbrication est zéro pour les tables qui sont appelées directement par le Générateur.

Le nombre d'espaces entre le caractère > et l'instruction **CALL** est important parce que le mécanisme d'alignement ajoute ce nombre d'espaces à la colonne de départ de chaque ligne générée par la table appelée. Seules les lignes de commentaire ne suivent pas cette règle.

DUMP

L'instruction *DUMP*

Si le Générateur rencontre une instruction *DUMP*, les actions suivantes seront exécutées:

- Le contenu de toutes les variables MTL est écrit en code MGL.
- Le nom du dernier sous-programme appelé dans le Générateur sera affiché dans le code MGL.
- Le contenu des variables du Générateur sera listé dans le fichier MGL.
- Le processus de génération est arrêté, mais le code MGL ne sera pas supprimé.
- La sortie MGL contient un message dans le format suivant:

```
\TBG568E 1 ** DUMP STATEMENT EXECUTED ** GENERATION INTERRUPTED **
```

Syntaxe:

>DUMP

Résultats:

Le code Source COBOL contiendra une section dans le format suivant:

```
***** DUMP OUTPUT *****
SYS-GENERATOR-VERSION = '90201'
SYS-OPERATING-SYSTEM = 'MVS'
SYS-COBOL-VERSION = '085'
SYS-COBOL = 'VS'
SYS-COMMA = ','
SYS-DECIMAL-POINT = '.'
SYS-NULL = ''
SYS-QUOTE = '"'
SYS-DATE-FORMAT = 'ISO'
SYS-NULABLE = 'N'
SYS-SQL-DIALECT = 0
SYS-EXEC = 'NULL'
SYS-DATE-WRITTEN = 20020122
SYS-PROTECTION-LIMIT = 14
SYS-WORK-SPACE = 'EX0'
*****
TABLE 400000 WAS DRIVEN BY PROGRAM PPTGGS
*****
***** TABLE SPECIFIC VARIABLES *****
#1 = 'EX0'
#2 = ' '
#3 = '20020122173850'
#4 = '0001'
#6 = 'MINERVA INTERNAL'
#7 = '6497970164970144083868'
#8 = 'QUOTE'
#9 = '07 01 06 BUILD 03'
#A = '07 01 06 BUILD 01'
*****
```

EXIT Table

La commande *EXIT* permet de quitter une table:

- pour retourner à table appelante, si le niveau d'imbrication n'est pas égal à zéro. Pour l'instruction *EXIT*, les mêmes règles s'appliquent que pour l'instruction *GOTO*.
- Pour continuer avec la table suivante (si le niveau d'imbrication est égal à zéro).

Syntaxe:

>EXIT

Règle:

Une instruction *EXIT* peut s'effectuer à partir de l'intérieur d'un bloc *IF* ou *EVALUATE*.

FREEZE et UNFREEZE

La commande *FREEZE* arrête l'interpréteur de la table de code (dont fait partie l'interpréteur MTL) de:

- modifier les guillemets dans la table de code en les guillemets standard.

Syntaxe:>FREEZE QUOTE

- modifier les variables spécifiques à la table (#1, #2, #A, etc...), pour que les développeurs de tables de code puissent laisser les caractères "#" tels quels.

Syntaxe:>FREEZE TS-VARS

- modifier les variables MTL pour que les développeurs puissent laisser les caractères "[" t "]" tels quels.
Syntaxe: >FREEZE MTL-VARS
- modifier l'indentation dans la table de code en l'indentation standard.
L'indentation du code généré sera automatiquement définie par la position du texte dans la table, ainsi que par la position de l'instruction CALL dans le cas d'une table appelée. La commande FREEZE arrête ce comportement standard et elle utilisera les paramètres d'indentation standard de la table.

Syntaxe: >FREEZE INDENT

UNFREEZE redémarre le traitement des lignes de code normal:

Syntaxe:

- >UNFREEZE QUOTE
- >UNFREEZE TS-VARS
- >UNFREEZE MTL-VARS
- >UNFREEZE INDENT

Exemples:

```
MOVE 'NO FREEZE QUOTE STATE' TO SENTENCE.
>FREEZE QUOTE
MOVE 'FREEZE QUOTE STATE' TO SENTENCE.
>UNFREEZE QUOTE
MOVE 'AGAIN NO FREEZE QUOTE STATE' TO SENTENCE.
```

Ceci engendre la codification générée suivante:

```
MOVE 'NO FREEZE QUOTE STATE' TO SENTENCE.
MOVE 'FREEZE QUOTE STATE' TO SENTENCE.
MOVE 'AGAIN NO FREEZE QUOTE STATE' TO SENTENCE.
```

GOTO

La commande de saut GOTO permet de sauter ou de répéter certaines commandes MTL et certaine codification COBOL dans le code MGL généré.

Syntaxe:

>GOTO { Label | Numéro de rangée }

Où:

- **GOTO:**

L'instruction **GOTO** est utilisée pour sauter vers une autre ligne DANS la table actuelle. Tout d'abord, le Générateur recherchera le label ou le numéro de ligne dans les lignes suivant la ligne courante. Si le label ou le numéro de ligne ne sont pas trouvés, la table sera relue et la recherche continue. Si le label ou le numéro de ligne ne sont pas trouvés la deuxième fois, une erreur sera générée.

- **label:**

Les labels sont des indicateurs marquant un point spécifique dans une table. Un label peut être tout type d'instruction MTL qui n'est pas une commande connue. Il peut contenir un à huit caractères. Si un label contient plus de huit caractères, seuls les huit premiers caractères seront utilisés. Les labels peuvent remplacer les numéros de ligne dans les instructions de saut.

- **Numéro de rangée:**

Les numéros de rangée sont représentés par 4 chiffres de 0001 à 9999. Vous pouvez découvrir le numéro de ligne d'une certaine instruction dans une table en utilisant l'instruction *LIST TABLE*.

Limitations:

La commande **GOTO** peut uniquement être utilisée:

- de l'intérieur d'un bloc IF ou EVALUATE vers l'extérieur de ce bloc
- de l'intérieur d'une boucle FOR-NEXT vers cette même boucle FOR-NEXT

Un saut ne peut jamais être effectué de l'extérieur d'une structure vers l'intérieur de cette structure.

Exemple:

```
>REMARK A GOTO CAN BE DONE UPWARDS
>REMARK BUT SEARCHES THE LABEL WITHIN THE TABLE
>GOTO LABEL2
)
* THIS LINE WILL BE SKIPPED
>LABEL2
* THIS IS THE LINE AFTER THE LABEL2 LABEL
*
* LOOP EXAMPLE *
>SET NUMBER=10
A01 TABLE-X
B02 TABLE-X-ELEMENT OCCURS [NUMBER] TIMES
B PIC X(15)
A01 REDEFINES TABLE-X
>SET COUNTER=0
)LABEL0
)IF COUNTER <> [NUMBER]
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '[COUNTER]' *
> SET COUNTER=[COUNTER]+1
B 02 TABLE-ELEMENT-[COUNTER] PIC X(15)
> GOTO LABEL0
>END-IF
```

Résultat:

```
* THIS IS THE LINE AFTER THE LABEL2 LABEL
*
* LOOP EXAMPLE *
01 TABLE-X
02 TABLE-X-ELEMENT OCCURS 10 TIMES PIC X(15)
01 REDEFINES TABLE-X
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '0' *
02 TABLE-ELEMENT-1 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '1' *
02 TABLE-ELEMENT-2 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '2' *
02 TABLE-ELEMENT-3 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '3' *
02 TABLE-ELEMENT-4 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '4' *
02 TABLE-ELEMENT-5 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '5' *
02 TABLE-ELEMENT-6 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '6' *
02 TABLE-ELEMENT-7 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '7' *
02 TABLE-ELEMENT-8 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '8' *
02 TABLE-ELEMENT-9 PIC X(15)
* COUNTER IS NOT EQUAL TO 10 BECAUSE IT IS '9' *
02 TABLE-ELEMENT-10 PIC X(15)
```

IF imbriqué

Il est possible d'imbriquer des blocs IF et EVALUATE multiples. Jusqu'à 16 niveaux d'imbrication sont autorisés.

Exemple:

```
>SET X=5
>SET Y=6
>IF X > Y
>THEN SET M='X IS BIGGER THAN Y'
* X>Y *
>ELSE IF X = Y
> THEN SET M='X EQUALS Y'
* X=Y *
> ELSE SET M='Y IS BIGGER THAN X'
* X<Y *
> END-IF
>END-IF
* MESSAGE : [M] *
```

Résultat:

```
* X<Y *
* MESSAGE : Y IS BIGGER THAN X *
```

* X<Y*

* MESSAGE: Y IS MORE THAN X*

Vu que X est établi à 5 et que Y est établi à 6, la première instruction **IF** sera fausse.

L'expression **ELSE** sera évaluée, et la deuxième instruction **IF** sera exécutée. Vu que *X* n'est pas égal à *Y*, la deuxième expression sera également fausse. La deuxième instruction **ELSE** sera utilisée et la variable *M* deviendra **Y IS BIGGER THAN X**.

IMPLEMENT

La commande *IMPLEMENT* est utilisée pour appeler une section à un stade ultérieur. Elle se souvient des tables déjà appelées et les appelle à la fin de la génération MGL.

Syntaxe:

>**IMPLEMENT** <tablename> [param1] [param2] [param3] [param4]

Résultat:

La table <**TABLENAME**> sera générée à la fin du MGL, en utilisant les paramètres [param1] [param2] [param3] [param4].

Dans la table, les paramètres peuvent être utilisés comme #1, #2, #3 et #4.

REMARK

La commande *REMARK* a un caractère purement informatif. Il offre à l'utilisateur la possibilité d'ajouter des commentaires dans la codification MTL.

Syntaxe:

>**REMARK THIS LINE WILL BE SHOWN WHEN TRACE IS ON (LOW)**

ou

>***THIS LINE WILL BE SHOWN WHEN TRACE IS ON (HIGH)**

Résultat:

Si vous générez cette codification en mode TRACE-ON, les lignes de remarque s'afficheront en mode commentaire.

```
>REMARK & SKIP CAN BE DONE DOWNWARDS BUT NOT UPWARDS
>REMARK SKIP IS THE ONLY STRUCTURE THAT CAN PASS
>REMARK THE BORDER OF A TABLE
>SKIP LABEL1
>REMARK THIS LINE WILL BE SKIPPED
```

SET

La commande *SET* crée ou modifie une variable MTL.

Syntaxe:

>**SET** Nom de la variable = Expression Nom de la variable

Où:

- **Nom de la variable** est le nom de la variable qui sera déclarée et qui obtiendra une valeur spécifique par la commande SET. Elle sera appelée la "variable SET".
- **Expression** est l'expression à inspecter. La valeur de résultat deviendra la valeur de la variable SET.

Exemple:

```
*****
**** WORKING WITH CHARACTER VARIABLES ****
*****
*
>SET AV0000-USER-NAME='*6'
>SET AV0000-PROGRAM-ID='*1'
>SET MESSAGE1='*6 HAS WRITTEN *1 ON *3'
>SET MESSAGE2='[AV0000-USER-NAME] HAS WRITTEN '
>SET MESSAGE2='[MESSAGE2][AV0000-PROGRAM-ID] ON '
>SET MESSAGE2='[MESSAGE2][SYS-DATE-WRITTEN]'

* MESSAGE1 = '[MESSAGE1]'
* MESSAGE2 = '[MESSAGE2]'

>SET VOORDQ='AZERTY'BZERTY'
>SET VOORDQ='AZERTY'BZERTY'

* VOORDQ = [[VOORDQ]] & VOORDQ = [[VOORDQ]] *
*
*****
*** WORKING WITH NUMERIC VARIABLES ***
*** SINCE NUMERIC EXPRESSIONS CAN ONLY CONTAIN ***
*** NUMERIC OPERATORS AND NUMERIC CHARACTERS ***
*** WE'LL HAVE TO WORK WITH THE VARIABLE CONTENTS ***
*****
*
>SET X=5
>SET Y=X
>SET Z=[X]+4
>SET Y=[Y]-2*[X]
>SET Z=[X]+10*[Y]/[X]
* X=[X], Y=[Y], Z=[Z] *
```

Résultat:

```
*****
**** WORKING WITH CHARACTER VARIABLES ****
*****
*
* MESSAGE1 = '[MESSAGE1]'
* MESSAGE2 = '[MESSAGE2]'
* VOORDQ = [AZERTY'BZERTY] & VOORDQ = [AZERTY'BZERTY] *
*
*****
*** WORKING WITH NUMERIC VARIABLES ***
*** SINCE NUMERIC EXPRESSIONS CAN ONLY CONTAIN ***
*** NUMERIC OPERATORS AND NUMERIC CHARACTERS ***
*** WE'LL HAVE TO WORK WITH THE VARIABLE CONTENTS ***
*****
*
* X=9, Y=-13, Z=-5 *
```

SKIP

La commande de saut *SKIP* permet de sauter ou de répéter certaines commandes MTL et certaine codification COBOL dans le code MGL généré.

Syntaxe:

>SKIP { Label | Numéro de rangée }

Où:

- **SKIP:**

L'instruction **SKIP** est utilisée pour sauter vers une autre ligne **après** la ligne courante de la table courante. Tout d'abord, le Générateur recherchera le label ou le numéro de ligne dans les lignes suivant la ligne courante. Si le label ou le numéro de ligne ne sont pas trouvés, la table suivante sera lue et la recherche continue. Si le label ou le numéro de ligne ne sont pas trouvés après avoir lu et cherché dans toutes les tables suivantes, une erreur sera générée. Une instruction **SKIP** peut **dépasser une seule table**!

- **label:**

Les labels sont des indicateurs marquant un point spécifique dans une table. Un label peut être tout type d'instruction MTL qui n'est pas une commande connue. Il peut contenir un à huit caractères. Si un label contient plus de huit caractères, seuls les huit premiers caractères seront utilisés. Les labels peuvent remplacer les numéros de ligne dans les instructions de saut.

- **Numéro de rangée:**

Les numéros de rangée sont représentés par 4 chiffres de 0001 à 9999. Vous pouvez découvrir le numéro de ligne d'une certaine instruction dans une table en utilisant l'instruction *LIST TABLE*.

Exemple:

```
>REMARK & SKIP CAN BE DONE DOWNWARDS BUT NOT UPWARDS
>REMARK SKIP IS THE ONLY STRUCTURE THAT CAN PASS
>REMARK THE BORDER OF A TAELE
>SKIP LABEL1
* THIS LINE WILL BE SKIPPED
)LABEL1
* THIS IS THE LINE AFTER THE LABEL1 LABEL
* THIS COMMAND WILL SKIP THE REST OF CURRENT TAELE
>SKIP 0001
* THIS LINE WILL NOT BE GENERATED
```

Résultats:

```
* THIS IS THE LINE AFTER THE LABEL1 LABEL
* THIS COMMAND WILL SKIP THE REST OF CURRENT TAELE
```

Structures EVALUATE

Les structures *EVALUATE* sont utilisées si l'action à effectuer dépend de plusieurs valeurs de variable. L'instruction *EVALUATE* évalue une expression et exécute les instructions correspondant au résultat de l'évaluation.

Syntaxe:

> EVALUATE Expression

[> WHEN Expression

[> THRU Expression]]°

[... instruction ...]¹

[> WHEN OTHER

[... instruction ...]°]

> END-EVALUATE

Où:

- **EVALUATE:**

EVALUATE est la première instruction MTL dans le bloc **EVALUATE**. Elle est suivie d'une expression MTL. Le contenu de cette expression est calculé. Par la suite, le résultat et le type de résultat (numérique ou alphanumérique) seront sauvegardés en mémoire.

- **Expression:**

expression se réfère à une expression MTL régulière.

- **Instruction:**

Instruction se réfère au code COBOL ou à l'instruction MTL.

- **WHEN:**

Plusieurs instructions **WHEN** peuvent être mises dans un bloc **EVALUATE**. Une instruction **WHEN** est suivie d'une expression MTL. Le contenu de cette expression est calculé. Par la suite, le résultat et le type de résultat (numérique ou alphanumérique) seront comparés au résultat de l'expression **EVALUATE**. Si le résultat est identique, les instructions suivant l'instruction **WHEN** seront effectuées.

Exemple:

```
> EVALUATE VAR1
> WHEN 'A'
* 'VAR1 IS 'A''
> WHEN 'C'
* 'VAR1 IS 'C''
> END-EVALUATE
```

- **THRU:**

Une instruction **WHEN** peut être suivie d'une instruction **THRU**. Cela signifie que toutes les valeurs d'expression entre la valeur de l'expression **WHEN** sur la ligne précédente et la valeur de l'expression

THRU sur cette ligne seront comparées avec la valeur de l'expression **EVALUATE**.

Exemple:

```
> EVALUATE VAR1
> WHEN 'A'
> THRU 'C'
> SET VAR2 = 'VAR1 IS 'A' OR 'B' OR 'C'
> WHEN 'E'
> THRU 'G'
> SET VAR2 = 'VAR1 IS 'E' OR 'F' OR 'G'
> END-EVALUATE
```

- **WHEN multiple**

Plusieurs instructions **WHEN-THRU** peuvent être utilisées l'une après l'autre. Dans ce cas, elles seront combinées.

Exemple:

```
> EVALUATE VAR1
> WHEN 'A'
> WHEN 'B'
> SET VAR2 = 'VAR1 IS 'A' OR 'B'
> WHEN 'C'
> WHEN 'G'
> THRU 'K'
> SET VAR2 = 'VAR1 IS IN 'OGHIJK'
> END-EVALUATE
```

- **WHEN OTHER:**

L'instruction **WHEN OTHER** est utilisée pour indiquer toutes les valeurs d'expression qui ne sont pas couvertes par des instructions **WHEN-THRU** antérieures dans le bloc **EVALUATE** courant.

Exemple:

```
> EVALUATE VAR1
> WHEN 'C'
> WHEN 'E'
> THRU 'H'
> SET VAR2 = 'VAR1 IS IN 'CEFGH'
> WHEN OTHER
> SET VAR2 = 'VAR1 IS INVALID !'
> END-EVALUATE
```

- **END-EVALUATE:**

Cette commande termine le bloc d'instruction **EVALUATE**.

Note: L'imbrication de multiples blocs **IF** et **EVALUATE** est possible. Jusqu'à 16 niveaux d'imbrication sont autorisés.

Structures IF

Les structures *IF* sont nécessaires pour exécuter des actions à condition.

Syntaxe:

```
>IF Expression Opérateur de comparaison Expression
[{ >OR Expression Opérateur de comparaison Expression
| >AND Expression Opérateur de comparaison Expression }]°
[ > THEN ] [... Instruction MTL ...]
[ ... instruction ...]°
[ > ELSE [... Instruction MTL ...]
[... instruction ...]° ]
> END-IF
```

Où:

- **Opérateur de comparaison** est un des opérateurs suivants:
= est égal à

<> n'est pas égal à
 < est inférieur à
 > est supérieur à
 >= n'est pas inférieur à
 <= n'est pas supérieur à

- **Expression** est une expression MTL régulière
- Une **Instruction** est un morceau de code COBOL ou une instruction MTL.
- Une **Instruction MTL** est une instruction MTL ou pas d'instruction.
- **IF/AND/OR** sont des opérateurs booléens utilisés pour spécifier la condition pour laquelle certaines actions doivent être effectuées. Des combinaisons AND-OR multiples peuvent suivre une instruction IF.

AND est plus fort que *OR*:

IF condition1 AND condition2 OR condition3 signifie que la condition1 ainsi que la condition2 doivent être remplies, ou, si cela n'est pas le cas, que la condition3 doit être remplie.

Exemple:

```

> IF MIN < MAX
> AND X > [MIN] + 1
> AND X < [MAX] - 1
> THEN SET COM = 'X IS STRICTLY BETWEEN [MIN] AND [MAX]'
B SUBTRACT [X] FROM F#1-R#1-RECORD-SIZE
> ELSE SET COM = 'X IS NOT STRICTLY BETWEEN [MIN] AND [MAX]'
B ADD 1 TO F#1-R#1-RECORD-SIZE
> END-IF
    
```

• THEN

La commande **THEN** suit la combinaison IF-AND-OR. Elle peut être suivie d'une commande MTL sur la même ligne. Les lignes qui suivent la commande **THEN** peuvent également être des commandes MTL, ou pourraient être des lignes COBOL qui doivent être écrites dans le programme COBOL généré.

Toutes les instructions entre **THEN** et **END-IF**, ou entre **THEN** et **ELSE** seront exécutées si la combinaison **IF-AND-OR** est vraie.

• ELSE

La commande **ELSE** est à l'opposé de la commande **THEN**: les instructions entre **ELSE** et **END-IF** seront exécutées si la combinaison **IF-AND-OR** est fausse. **ELSE** peut être suivi d'une commande MTL sur la même ligne.

• END-IF

END-IF termine le bloc d'instruction IF.

TRACE

La commande **TRACE** est utilisée pour tracer ou déboguer la codification MTL. Elle fournit plus d'informations concernant la logique appliquée par l'interpréteur MTL.

Vous pouvez utiliser la commande **TRACE** en combinaison avec les options suivantes:

- **OFF**: aucune phrase MTL n'apparaîtra dans le MGL/MRL généré.
- **ON** ou **LOW**: seules les instructions MTL pertinentes seront affichées comme commentaire.
Par exemple: dans une structure IF-THEN-ELSE, seule la branche suivant **IF** ou **ELSE** sera affichée dans le MGL ou MRL généré.
- **MEDIUM**: Les branches conditionnelles seront affichées dans les commentaires, mais les instructions MTL à cause d'instructions de saut (**GOTO** et **SKIP**) ne seront pas affichées.
- **HIGH**: Chaque ligne MTL sera placée comme commentaire dans le MGL ou le MRL, y comprises les lignes ignorées à cause une instruction de saut. Même les remarques commençant par ">*" doivent être mis

en MGL ou MRL.

Syntaxe:

>TRACE { OFF | ON | LOW | MEDIUM | HIGH }

Exemples:

Code d'exemple:

```
>TRACE LOW
>SET X=5
>IF X=3
>  REMARK X IS THREE
>ELSE
>  REMARK X IS SOMETHING ELSE THAN THREE
>  GOTO THE-END
>END-IF
>REMARK NOT THE END
>THE-END
>REMARK THIS IS THE END
```

Résultat:

Si TRACE = LOW

```
*TRACE LOW
*SET X=5
01 T *ELSE
01 T *  REMARK X IS SOMETHING ELSE THAN THREE
    *THE-END
    *REMARK THIS IS THE END
```

Si TRACE = MEDIUM

```
*TRACE MEDIUM
*SET X=5
01 F *IF X=3
01 F *  REMARK X IS THREE
01 T *ELSE
01 T *  REMARK X IS SOMETHING ELSE THAN THREE
    *THE-END
    *REMARK THIS IS THE END
```

Si TRACE = HIGH

```
*TRACE HIGH
*SET X=5
01 F *IF X=3
01 F *  REMARK X IS THREE
01 T *ELSE
01 T *  REMARK X IS SOMETHING ELSE THAN THREE
    G  *  GOTO THE-END
    G  *END-IF
    G  *REMARK NOT THE END
    G  *THE-END
    *REMARK THIS IS THE END
```

Remarques:

- Les lignes qui n'ont pas été interprétées dans une instruction conditionnelle sont précédées par la lettre *F* (pour *False (Fausse)*) dans la colonne 4.
- Les lignes qui n'ont été interprétées dans une instruction conditionnelle sont précédées par la lettre *T* (pour *True (Vraie)*) dans la colonne 4.
- Lors de l'interprétation des instructions conditionnelles, le niveau d'imbrication est affichée dans les colonnes une et deux.
- Les lignes qui n'ont pas été interprétées à cause d'une instruction de saut sont précédées par la lettre "G" dans le cas d'une instruction GOTO, et par la lettre "S" dans le cas d'une instruction SKIP. Ces caractères sont placés dans la colonne 3.

Traitement des messages d'erreur

Dans le Langage de modèle MetaSuite vous pouvez programmer vos propres messages d'erreur et arrêter le processus de génération, si nécessaire.

Il existe quatre types de messages d'erreur:

- **ERROR:**
Ces messages indiquent des erreurs graves qui arrêtent le processus de génération.
- **WARNING:**
Ces messages indiquent des erreurs moins graves qui n'arrêtent pas le processus de génération.
- **MESSAGE:**
Ces messages informent l'Utilisateur d'un évènement. Le processus de génération ne sera pas arrêté.
- **USER:**
Il s'agit de messages informatifs définis par l'Utilisateur. Ils n'ont rien à voir avec la table MESSAGE dans le Dictionnaire MetaSuite.

Syntaxe:

>{ERROR|WARNING|MESSAGE} [Numéro de message][Texte du message]

Où:

- **ERROR, WARNING ou MESSAGE:**
Le type de Message d'erreur requis.
- **Numéro de message:**
Saisissez le numéro correspondant au message requis dans la table de messages du Dictionnaire (AU0001). Pour plus d'informations concernant les erreurs du Générateur, se référer à la section appropriée. Si vous omettez ce numéro, un numéro d'erreur par défaut sera généré.
- **Texte du message:**
Vous pouvez saisir du texte libre. Si non, un texte par défaut sera affiché.

Le tableau suivant donne un aperçu des formats possibles de la commande **ERROR** et des messages résultants dans la sortie générée:

Commande	Résultats générés
>ERROR	\TBG570E 1 GENERAL MTL MESSAGE (TABLE AV0000 LINE 0026) Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • E = Erreur • 1 = numéro séquentiel des erreurs sur cette ligne • GENERAL MTL ERROR = Message d'erreur par défaut • TABLE AV0000 LINE 0026 = la table et la ligne où l'erreur est rencontrée.
>ERROR Texte du message	\TBG570E 1 GENERAL MTL ERROR TEST123 Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • E = Erreur • 1 = numéro séquentiel des erreurs sur cette ligne • GENERAL MTL ERROR = Message d'erreur par défaut • TEST123 = Texte de message défini par l'Utilisateur

Commande	Résultats générés
>ERROR Numéro de message	\TBG570E 1 SPACE NOT ALLOWED (TABLE AV0000 LINE 0026) Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • E = Erreur • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message d'erreur correspondant au numéro tel que disponible dans la table de messages par défaut AU0001 • TABLE AV0000 LINE 0026 = la table et la ligne où l'erreur est rencontrée.
>ERROR Numéro de message Texte du message	\TBG570E 1 SPACE NOT ALLOWED TEST123 Où: • TBG570 = Le module du Générateur MetaSuite générant l'erreur • E = Erreur • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message d'erreur correspondant au numéro tel que disponible dans la table de messages par défaut AU0001 • TEST123 = Texte de message défini par l'Utilisateur

Le tableau suivant donne un aperçu des formats possibles de la commande **WARNING** et des messages résultants dans la sortie générée:

Commande	Résultats générés
>WARNING	\TBG570W 1 GENERAL MTL MESSAGE (TABLE AV0000 LINE 0026) Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • W = Avertissement (Warning) • 1 = numéro séquentiel des erreurs sur cette ligne • GENERAL MTL MESSAGE = Message d'avertissement par défaut • TABLE AV0000 LINE 0026 = la table et la ligne où l'erreur est rencontrée.
>WARNING Texte du message	\TBG570W 1 GENERAL MTL MESSAGE TEST123 Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • W = Avertissement (Warning) • 1 = numéro séquentiel des erreurs sur cette ligne • GENERAL MTL MESSAGE = Message d'avertissement par défaut • TEST123 = Texte de message défini par l'Utilisateur
>WARNING Numéro de message	\TBG570W 1 SPACE NOT ALLOWED (TABLE AV0000 LINE 0026) Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • W = Avertissement (Warning) • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message d'avertissement correspondant au numéro tel que disponible dans la table de messages par défaut AU0001 • TABLE AV0000 LINE 0026 = la table et la ligne où l'erreur est rencontrée.

Commande	Résultats générés
>WARNING Numéro de message Texte du message	\TBG570W 1 SPACE NOT ALLOWED TEST123 Où: • TBG = Le module du Générateur MetaSuite générant l'erreur • 570 = numéro d'erreur • W = Avertissement (Warning) • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message d'avertissement correspondant au numéro tel que disponible dans la table de messages par défaut AU0001 • TEST123 = Texte de message défini par l'Utilisateur

Le tableau suivant donne un aperçu des formats possibles de la commande **MESSAGE** et des messages résultants dans la sortie générée:

Commande	Résultats générés
>MESSAGE	\
>MESSAGE Texte du message	\THIS IS A USER MESSAGE Où: • THIS IS A USER MESSAGE = le texte du message défini par l'Utilisateur
>MESSAGE Numéro de message	\TBG125 1 SPACE NOT ALLOWED (TABLE AV0000 LINE 0026) Où: • 125 = le Numéro de message indiqué • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message correspondant au numéro de message tel que disponible dans la table de messages par défaut AU0001
>MESSAGE Numéro de message Texte du message	\TBG125 1 SPACE NOT ALLOWED TEST123 Où: • 125 = le Numéro de message indiqué • 1 = numéro séquentiel des erreurs sur cette ligne • SPACE NOT ALLOWED = un message d'avertissement correspondant au numéro tel que disponible dans la table de messages par défaut AU0001 • TEST123 = Texte de message défini par l'Utilisateur

UNSET

La commande *UNSET* retire une variable MTL.

Syntaxe:

>UNSET Nom de la variable

Où:

- **Nom de la variable** est le nom de la variable à retirer.

Exemple:

```
*****
**** WORKING WITH SET AND UNSET ****
*****
>SET TEST=1
>SET X=FUNCTION IS-SET(TEST)
*TEST=[TEST] AND X=[X]
>UNSET TEST
>SET X=FUNCTION IS-SET(TEST)
*TEST=[TEST] AND X=[X]
```

Résultat:

```
*****
**** WORKING WITH SET AND UNSET ****
*****
*TEST=1 AND X=TRUE
*TEST=0 AND X=FALSE
```

15.4. Variables système MTL

Dans le Langage de modèle MetaSuite, certaines variables système sont prédéfinies. Ces variables sont appelées "protégées" parce qu'elles ne peuvent pas être modifiées par la commande SET. Elles contiennent les valeurs définies dans le Dictionnaire MetaSuite.

Le tableau suivant décrit ces variables MTL système

Variable	Description	Exemple
SYS-CALL-DYNAMIC	Cette option est établie à Y si le code COBOL généré utilise des appels COBOL dynamiques pour invoquer les modules d'exécution. Elle est établie à N, si les appels COBOL ne sont pas dynamiques.	Y or N
SYS-COBOL	Il s'agit de la version du compilateur COBOL pour lequel le code MGL sera généré, représenté comme une chaîne de caractères à 2 chiffres.	VS
SYS-COBOL-VERSION	Il s'agit de la version du compilateur COBOL pour lequel le code MGL sera généré, représenté comme un numéro à 3 chiffres.	085
SYS-DATE-FORMAT	Le paramètre du format de date courant.	ISO, EUR ou JIS
SYS-DATE-WRITTEN	La date de génération dans le format suivant: YYYYMMDD.	20130411
SYS-DECIMAL-POINT	Le paramètre de la décimale courante.	. ou ,
SYS-EXEC	Le paramètre EXEC courant.	IMS ou NULL
SYS-FIELD-COUNT	Le nombre de champs dans la table de "Champs". Cette table contient les Champs Cible, les Champs de travail et les Champs Source, dans cet ordre.	156
SYS-FILE-COUNT	Le nombre de Fichiers Source.	20
SYS-GENERATOR-VERSION	Il s'agit de la version courante du Générateur, représentée comme un numéro à 5 chiffres.	80103
SYS-MODEL-NAME	Le nom du Modèle MetaSuite.	Modèle d'exemple
SYS-NUL	Il s'agit du paramètre du caractère Null courant.	\
SYS-NULABLE	Le paramètre NULABLE initial.	N ou I (i majuscule)

Variable	Description	Exemple
SYS-OPERATING-SYSTEM	Il s'agit du Système d'exploitation pour lequel le code MGL sera généré, représenté comme une chaîne de caractères à 3 chiffres.	BS2
SYS-PROGRAM-NAME	Le nom du programme généré.	SampleModel.exe
SYS-PROTECTION-LIMIT	Le nombre de variables protégées, y compris celle-ci.	25
SYS-QUOTE	Le paramètre QUOTE initial.	' ou "
SYS-RECORD-COUNT	Le nombre d'Enregistrements Source.	24562
SYS-SQL-DIALECT	Il s'agit du paramètre de dialecte SQL courant, représenté par une valeur entre 1 et 15.	0 = Non spécifié 1 = DB2 sous z/OS 2 = DB2 sous DOS/VSE 3 = DB2/2 4 = DB2 sous OS/400 5 = DB2 LUW 6 = Oracle 7 = Ingres 8 = Sybase 9 = SQL Server 10 = Informix 11 = SESAM 12 = Oracle/RDB 13 = ODBC 14 = Teradata 15 = MySQL
SYS-USER	Le nom de l'Utilisateur qui a généré ce programme.	john
SYS-SQL-FOUND	Si le code MXL généré contient des options liées à SQL, cette variable reçoit la valeur "TRUE". Si non: "FALSE".	
SYS-INTERPUNCTION	Contient le paramètre INTERPUNCTION par défaut. Sa valeur peut être "NO", "YES" ou "OFF".	
SYS-TARGET-COUNT	Le nombre de Fichiers Cible.	
SYS-RECORD-COUNT	Le nombre d'Enregistrements Source.	
SYS-DETAIL-COUNT	Le nombre d'Enregistrements Cible.	
SYS-FIRST-FIELD	Le numéro symbolique de premier Champ Source dans la table des Champs.	
SYS-UNICODE	Si Unicode est supporté sur votre système, la valeur sera établie à "TRUE".	
SYS-ONE-CCSID	Si un seul CCSID est utilisé dans le Fichier MXL, la valeur "TRUE" sera attribuée à cette variable. Si non: "FALSE".	
DEFAULT-CCSID	Cette variable numérique contient le CCSID par défaut pour les caractères.	

Variable	Description	Exemple
UNICODE-CCSID	Cette variable numérique contient le CCSID par défaut pour Unicode.	
SYS-DEBUG	Établie à "TRUE" si la génération est effectuée en utilisant l'option DEBUG.	
SYS-NATIONAL-IS-USED	Si le code MXL contient des Champs Unicode, cette variable reçoit la valeur "TRUE". Si non: "FALSE".	

15.5. Variables MTL spécifiques au Fichier Source

Le Générateur MetaSuite offre au programmeur certaines variables spécifiques au Fichier. Ces variables sont en sous-script ou indexées. L'index est un numéro de Fichier interne. Cet index peut avoir une valeur entre 1 et SYS-FILE-COUNT.

Le tableau suivant décrit ces variables MTL spécifiques aux champs

Variable	Description
FILE-DBNAME (...)	Le nom de la base de données.
FILE-LENGTH (...)	La longueur du Fichier. Cette longueur est l'espace qui sera utilisé par le programme COBOL généré pour définir le bloc de contrôle de Fichier. Elle est au moins égale à la taille du plus grand Enregistrement dans ce Fichier.
FILE-NAME (...)	Le nom du Fichier.
FILE-RECORD-FORMAT (...)	Il s'agit du format d'Enregistrement tel que défini dans la base de données MetaStore. Les valeurs possibles sont: • FIXED • VARIABLE • UNDEFINED
FILE-RECORDING-MODE (...)	Il s'agit du Mode d'enregistrement tel que spécifié dans la base de données MetaStore. Les valeurs possibles sont: • ASCII • EBCDIC • NATIVE
FILE-TYPE (...)	il s'agit du type de Fichier indiquant quel Fichier est un Fichier de paramètres, une Matrice externe avec ou sans clé, etc.
FILE-CONTAINS-UCD (...)	"TRUE" ou "FALSE". "TRUE" si le Fichier contient des Champs Unicode/National.
FILE-IS-DELIMITED (...)	"TRUE" ou "FALSE". "TRUE" si le Fichier est un Fichier Source délimité.

Variable	Description
FILE-XPATH-COUNT (...)	Le nombre de niveaux XPATH dans la propriété XPATH du Fichier Source. Lors de l'émission de cette commande, le pointeur MTL XPATH est placé au départ du chemin (XPath) dans le Fichier Source. Par exemple: Si le XPATH du Fichier numéro 7 est '/XML/HEAD/CORNER', le FILE-XPATH-COUNT (7) sera établi à 3.
FILE-XPATH-NEXT (...)	Le niveau XPATH suivant dans la propriété XPATH du Fichier Source. Dans notre exemple précédent, FILE-XPATH-NEXT (7) sera 'XML'. La prochaine fois que FILE-XPATH-NEXT (7) sera généré, le résultat sera 'HEAD' parce que c'est la partie suivante (NEXT) du XPATH. À chaque exécution de cette commande, le pointeur XPATH avance jusqu'à ce qu'il atteigne la fin de la chaîne de caractères de l'XPath. À la fin, le résultat sera une chaîne de caractères vide.
FILE-FIRST-FIELD (...)	Le numéro de Champ symbolique le plus bas de tous les Champs qui appartiennent à ce Fichier. Cette fonction a été développée dans le but d'accélérer le code MTL.
FILE-NBR-RECORDS (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.
FILE-IS-TERMINATED (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.
FILE-IS-OBF (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.
FILE-OBF-NUMBER (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.

15.6. Variables MTL liées à l'Enregistrement Source

Le Générateur MetaSuite offre au programmeur certaines variables spécifiques à l'Enregistrement. Ces variables sont en sous-script ou indexées. L'index est un numéro d'Enregistrement interne. Cet index peut avoir une valeur entre 1 et SYS-RECORD-COUNT.

Le tableau suivant décrit ces variables MTL spécifiques aux enregistrements.

Variable	Description
RECORD-DBNAME (...)	Le nom de la base de données.
RECORD-FILE (...)	L'index du Fichier Source auquel appartient l'Enregistrement Source.
RECORD-LENGTH (...)	La longueur de l'Enregistrement.
RECORD-NAME (...)	Le nom de l'Enregistrement.
RECORD-OCCURRANCY (...)	Le numéro d'occurrence de l'Enregistrement spécifié dans le chemin.

Variable	Description
RECORD-XPATH-COUNT (...)	Le nombre de niveaux XPATH dans la propriété XPATH de l'Enregistrement Source. Lors de l'émission de cette commande, le pointeur MTL XPATH est placé au départ du chemin (XPATH) dans l'Enregistrement. Si le XPATH de l'Enregistrement Source est un chemin relatif, le XPATH du Fichier sera ajouté comme préfixe à ce XPATH. Par exemple: Supposons que l'Enregistrement 10 est un Enregistrement du Fichier 7. Si le XPATH du Fichier numéro 7 est '/XML/HEAD/CORNER' et que le XPATH de l'Enregistrement 10 est 'GOALS/POINTS', la valeur de RECORD-XPATH-COUNT (10) sera 5.
RECORD-XPATH-NEXT (...)	Le niveau XPATH suivant dans la propriété XPATH de l'Enregistrement. Dans notre exemple précédent, RECORD-XPATH-NEXT (10) sera 'XML'. La prochaine fois que RECORD-XPATH-NEXT (10) sera généré, le résultat sera 'HEAD' parce que c'est la partie suivante (NEXT) du XPATH entier. À chaque exécution de cette commande, le pointeur XPATH avance jusqu'à ce qu'il atteigne la fin de la chaîne de caractères de l'XPATH. À la fin, le résultat sera une chaîne de caractères vide.
RECORD-RAW-NAME (...)	Tandis que RECORD-NAME ne contient que des caractères en majuscules, RECORD-RAW-NAME contient l'Enregistrement original tel que spécifié dans le code MXL.
RECORD-COLSEP (..)	Séparateur de colonnes de l'Enregistrement Source.
RECORD-ROWTERM (..)	Le Code fin de rangée pour l'Enregistrement Source
RECORD-RCDCD-ELEM (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.
RECORD-RCDCD-ELEM-NXT (...)	Cette valeur ne sera plus prise en charge dans les versions futures. Ne peut être utilisé que par le service de Support de MetaSuite.

15.7. Variables MTL liées au Fichier Cible

Le Générateur MetaSuite offre au programmeur certaines variables spécifiques au Fichier. Ces variables sont en sous-script ou indexées. L'index est un numéro de Fichier Cible interne. Cet index peut avoir une valeur entre 1 et SYS-TARGET-COUNT.

Le tableau suivant décrit ces variables

Variable	Description
TARGET-XPAT-COUNT (...)	Le nombre de niveaux XPATH dans la propriété XPATH du Fichier Cible. Lors de l'émission de cette commande, le pointeur MTL XPATH est placé au départ du chemin (XPATH) dans le Fichier Cible. Par exemple: Si le XPATH du Fichier numéro 7 est '/XML/HEAD/CORNER', le FILE-XPAT-COUNT (7) sera établi à 3.

Variable	Description
TARGET-XPATH-NEXT (...)	Le niveau XPATH suivant dans la propriété XPATH du Fichier Cible. Dans notre exemple précédent, FILE-XPATN-NEXT (7) sera 'XML'. La prochaine fois que FILE-XPATN-NEXT (7) sera généré, le résultat sera 'HEAD' parce que c'est la partie suivante (NEXT) du XPATH. À chaque exécution de cette commande, le pointeur XPATH avance jusqu'à ce qu'il atteigne la fin de la chaîne de caractères de l'XPATH. À la fin, le résultat sera une chaîne de caractères vide.
TARGET-OUT-TYPE (...)	Le type de Fichier Cible. Le résultat peut être "T", "C", "D" pour les Fichiers Cible (Target), et "R" or "P" pour les Fichiers de Rapport.
TARGET-OUT-NUMBER (...)	Le numéro externe utilisé pour se référer à cette Cible, dans le code MXL ainsi que dans le code COBOL généré et les scripts.
TARGET-IN-NUMBER (xx)	Uniquement pour les Fichiers qui ne sont pas des Rapports: le numéro symbolique interne utilisé pour le Fichier Cible Txx. Il s'agit de la fonctionnalité inverse de la fonction précédente.
TARGET-XMLDECL-NBR (...)	Un Fichier XML commence par une phrase de déclaration XML. Cette phrase peut être une chaîne de caractères longue coupée en plus petits morceaux pour qu'ils rentrent dans la phrase COBOL. Le résultat de cette fonctionnalité est le nombre de parties dans lesquelles cette phrase sera coupée.
TARGET-XMLDECL-NXT (...)	Se référer également à TARGET-XMLDECL-NBR. Un Fichier XML commence par une phrase de déclaration XML. Cette phrase peut être une chaîne de caractères longue coupée en plus petits morceaux pour qu'ils rentrent dans la phrase COBOL. Cette fonction renvoie une partie de la déclaration XML. Pour un nouvel appel de fonction, une nouvelle partie de la déclaration XML sera renvoyée jusqu'à ce que la fin de la déclaration XML soit atteinte. Dans ce cas, une valeur vide sera renvoyée.
TARGET-CONTAINS-UNICODE (...)	Si un Fichier Cible contient des Champs Unicode, la valeur "TRUE" sera renvoyée, si non le résultat sera la valeur "FALSE".

15.8. Variables liées à l'Enregistrement Cible

Le Générateur MetaSuite offre au programmeur MTL certaines variables spécifiques au Fichier. Ces variables sont en sous-script ou indexées. L'index est le numéro d'Enregistrement Cible interne, ou le numéro de "détail". Cet index peut avoir une valeur entre 1 et SYS-DETAIL-COUNT. Le tableau suivant décrit ces variables spécifiques au Fichier MTL.

Le tableau suivant décrit ces variables

Variable	Description
DETAIL-NAME	Le nom de l'Enregistrement Cible.
DETAIL-TARGET	Le numéro symbolique du Fichier Cible auquel appartient cet Enregistrement.
DETAIL-LINE	Le Numéro de rangée détaillée, spécifié dans en MXL.
DETAIL-CCSID	Le CCSID de l'Enregistrement Cible.

Variable	Description
DETAIL-XPATH-COUNT	Le nombre de niveaux XPATH dans la propriété XPATH de l'Enregistrement Cible. Lors de l'émission de cette commande, le pointeur MTL XPATH est placé au départ du chemin (XPATH) dans l'Enregistrement. Si le XPATH de l'Enregistrement Cible est un chemin relatif, le XPATH du Fichier sera ajouté comme préfixe à ce XPATH. Par exemple: Supposons que l'Enregistrement 10 est un Enregistrement du Fichier 7. Si le XPATH du Fichier numéro 7 est '/XML/HEAD/CORNER' et que le XPATH de l'Enregistrement 10 est 'GOALS/POINTS', la valeur de RECORD-XPATH-COUNT (10) sera 5.
DETAIL-XPATH-NEXT	Le niveau XPATH suivant dans la propriété XPATH de l'Enregistrement Cible. Dans notre exemple précédent, RECORD-XPATH-NEXT (10) sera 'XML'. La prochaine fois que RECORD-XPATH-NEXT (10) sera généré, le résultat sera 'HEAD' parce que c'est la partie suivante (NEXT) du XPATH entier. À chaque exécution de cette commande, le pointeur XPATH avance jusqu'à ce qu'il atteigne la fin de la chaîne de caractères de l'XPATh. À la fin, le résultat sera une chaîne de caractères vide.
DETAIL-TYPE	Cette variable peut avoir 3 valeurs possibles: • "D" signifie Enregistrement de Détail. • "T" signifie TOTAL (Enregistrement de Total final) • "A" signifie ACCUMULATE (Enregistrement de groupe)

15.9. Variables MTL liées au Champ

Le Générateur MetaSuite offre au programmeur MTL certaines variables spécifiques au Champ. Ces variables sont en sous-script ou indexées. L'index est un numéro d'Enregistrement interne. Cet index peut avoir une valeur entre 1 et SYS-FIELD-COUNT.

Le tableau suivant décrit ces variables MTL spécifiques aux champs

Variable	Description
FIELD-DATE-TYPE (...)	Le type de date représenté comme un numéro. Si établi à 0, le type de date de ce champ n'est pas défini. Les types de données suivants sont disponibles: • 1 to #10 : Types NUMÉRIQUES. Cela signifie que la valeur numérique sera utilisée comme entrée. • 11 to #30 : Types ALPHANUMÉRIQUES. Cela signifie que la valeur alphanumérique sera utilisée comme entrée. • 11 to #20 : ALPHANUMÉRIQUE avec délimiteur • 21 to #30 : ALPHANUMÉRIQUE sans délimiteur • Ce format de date est déterminé par le dernier chiffre: - se terminant en 1: MMDDYY - se terminant en 2: DDMMYY - se terminant en 3: MMDDYYYY - se terminant en 4: DDMMYYYY - se terminant en 5: YYDDD - se terminant en 6: YYYYDDD - se terminant en 7: YYMMDD - se terminant en 8: YYDDMM - se terminant en 9: YYYYMMDD - se terminant en 0: YYYYDDMM
FIELD-DBNAME (...)	Le paramètre DB-NAME pour le Champ. Il peut également contenir une Valeur pour Null.
FIELD-DECIMALS (...)	Le nombre de décimales après le point décimal.
FIELD-FILE (...)	L'index du Fichier Source auquel appartient le Champ. Si le Champ n'appartient pas à un Fichier Source, la valeur est "0".
FIELD-GROUP (...)	Il s'agit du numéro de champ du Champ de groupe.
FIELD-SIZE-BYTES (...)	Le nombre d'octets réservés pour le Champ dans l'Enregistrement.
FIELD-NAME (...)	Le nom du Champ.
FIELD-OCCURS (...)	Le nombre total d'occurrences de ce Champ dans un seul Enregistrement. Ce numéro est obtenu en multipliant toutes les occurrences du niveau courant et des niveaux de regroupement plus élevés.
FIELD-OCCURS-LEVEL (...)	La valeur d'occurrence au niveau le plus bas.
FIELD-ORIGIN (...)	L'indication de l'origine du Champ. • G = Champ de groupe • F = Champ (Field) Source • W = Champ de travail (Work Field) • U = autre Champ
FIELD-POSITION (...)	Il s'agit de la position absolue du Champ dans l'Enregistrement.
FIELD-RAW-NAME (...)	Le nom du Champ en format RAW, incluant le suffixe <code>_nnnn#nnnnnn</code> .
FIELD-RECORD (...)	L'index de l'Enregistrement Source auquel appartient le Champ. La valeur est 0 si le Champ n'appartient pas au Fichier Source.
FIELD-SUBSCRIPT-LEVEL (...)	Le nombre de niveaux hiérarchiques qui se produisent plus d'une fois.

Variable	Description
FIELD-TYPE (...)	Le type de champ représenté comme un numéro.
FIELD-SEQUENCE (...)	Numéro de Séquence du Champ. Probablement, cette valeur ne sera plus prise en charge dans les versions futures de MetaSuite.
FIELD-MVOFLD (...)	Ne sera plus pris en charge par les versions ultérieures de MetaSuite. Ne peut être utilisé que par le service de Support de MetaSuite.
FIELD-DUMP (...)	Ne sera plus pris en charge par les versions ultérieures de MetaSuite.
FIELD-SIZE-BYTES (...)	La taille physique du Champ: le nombre d'octets.
FIELD-SIZE-CHARS (...)	La taille du Champ en nombre de caractères.
FIELD-IS-CODESET-SENSITIVE (...)	Certains types de Champ sont sensibles au jeu de codes, d'autres ne le sont pas. Si ce Champ est sensible à une modification du jeu de codes, la valeur renvoyée sera "TRUE", sinon elle sera "FALSE".
FIELD-IS-NUMERIC (...)	S'il s'agit d'un Champ numérique, la valeur retournée sera "TRUE", sinon elle sera "FALSE".
FIELD-IS-SIGNED (...)	S'il s'agit d'un Champ signé, la valeur retournée sera "TRUE", sinon elle sera "FALSE".
FIELD-MULT (...)	Renvoie une valeur de 10 caractères, contenant les informations suivantes: • Si le Champ est à occurrence multiple, le premier octet contient la valeur "Y". • Le deuxième octet contient le nombre de niveaux éventuels. • Octets 3 à 6: le nombre d'occurrences au premier niveau • Octets 7 à 10: le nombre d'occurrences au deuxième niveau
FIELD-NRINDEX (...)	Le nombre d'index qui peuvent être utilisés pour ce Champ.
FIELD-XMLNAME (...)	Le nom XML du Champ. Si la propriété XMLNAME n'est pas vide, cette valeur sera utilisée. Si la propriété XMLNAME n'existe pas, le nom du Champ sans préfixe ni suffixe sera utilisé.
FIELD-XMLTYPE (...)	Le type XML du Champ: attribut (valeur "A"), Champ de groupe (valeur "G") ou nœud (espace).
FIELD-XMLMASK (...)	Le masque d'édition XML du Champ.
FIELD-IS-ATTRIBUTE (...)	Établi à "TRUE" si FIELD-XMLTYPE (...) = "A". Si non, il est "FALSE".
FIELD-HAS-ATTRIBUTES (...)	Établi à "TRUE" s'il s'agit d'un Champ de groupe, et si au moins un des Sous-champs est un attribut. Si non, il est "FALSE".
FIELD-IS-GROUP (...)	Établi à "TRUE" si FIELD-XMLTYPE (...) = "G". Si non, il est "FALSE".
FIELD-IS-NULLABLE (...)	Établi à "TRUE" si le Champ est de type INNULL, OUTNULR, OUTNULL, ou s'il est de type DEFAULT et que DEFAULT-NULLABLE est "INNULL". En d'autres termes, cette variable indique si le Champ est Nullable ou pas.

Variable	Description
FIELD-INIT (...)	Si un Champ contient une valeur initiale, cette fonction renvoie la valeur "TRUE", sinon le résultat sera "DEFAULT". Exception: si un champ a un champ de groupe, et que ce champ de groupe a une valeur initiale, la valeur du champ de groupe a priorité sur la valeur du sous-champ. Dans ce cas, la fonction renvoie la valeur "FALSE".
FIELD-CCSID (...)	Le CCSID au niveau du champ. CCSID est l'abréviation de "Coded Character Set Identifier". Il s'agit d'un chiffre de 16 octets représentant une codification spécifique d'une page de code spécifique. Ce CCSID peut être spécifié au niveau général de MetaSuite (les paramètres de Dictionnaire CHARACTER-CCSID et UNICODE-CCSID), ainsi qu'au niveau du fichier, de l'enregistrement ou du champ.
FIELD-CCSID-TYPE (...)	FIELD-CCSID-TYPE est 1 pour les CCSIDs de type caractère, et 2 pour les CCSIDs Unicode.
FIELD-CCSID-LEN1 (...)	Le nombre de caractères que ce Champ comprend.
FIELD-CCSID-LEN2 (...)	La taille qu'un Champ Unicode devrait avoir si ce Champ était transféré vers un Champ Unicode. (Dans la plupart des cas: FIELD-CCSID-LEN1 (...) * 2)
FIELD-IS-TYPNAT (...)	Établi à "TRUE" si le type de Champ est "NATIONAL".
FIELD-SUBSCRIPT-LEVEL (...)	Le nombre de sous-scripts requis par ce Champ.
FIELD-SUBSCRIPT-HOLDER (...)	Pour les Champs élémentaires ou les Champs unidimensionnels: c'est le groupe qui contient l'index.
FIELD-GROUP (...)	Renvoie le numéro symbolique du Champ de groupe, ou un zéro s'il s'agit d'un Champ élémentaire.
FIELD-IS-1ST-KEY (...)	Renvoie la valeur "TRUEE" si ce Champ est le premier Champ clé.

Langage d'exécution MetaSuite

Le Langage d'exécution MetaSuite (MRL) est un ensemble de commandes ou d'instructions que vous pouvez utiliser pour exécuter vos programmes.

Les instructions MRL sont créées par le Générateur sur la base des tables de modèle.

16.1. Tables

Les tables suivantes sont utilisées par le Générateur MetaSuite pour créer les instructions MRL.

RL0000	En-tête du script #1 Nom du programme #2 Numéro de version du programme Cette table peut être utilisée pour supprimer les fichiers PPTLST et PPTLOG résultants d'une exécution précédente.
RL0001	Exécution du programme: #1 Nom du programme #2 Numéro de version du programme Cette table contient la commande pour exécuter le programme.
RL0002	Assignment du Fichier Source à la Matrice externe (PPTFnn): #1 Nom du programme #2 Numéro de matrice externe MetaSuite #3 Nom de matrice externe MetaSuite
RL0003	Assignment de Fichier Source (PPTFnn): #1 Nom du programme #2 Numéro de Fichier Source MetaSuite #3 Nom de Fichier Source MetaSuite
RL0004	Assignment de fichier temporaire (PPTTnn): #1 Nom du programme #2 Numéro du Fichier Source MetaSuite pour lequel un Fichier temporaire est nécessaire (dû à un Tri ou un Extrait de Fichier Source)
RL0005	Assignment du Fichier de données Cible (PPTTDnn): #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite #3 Taille de l'Enregistrement #4 Numéro de séquence du Fichier de sortie

RL0006	Assignment du format du Fichier Cible (PPTTFnn): #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite #3 Taille de l'Enregistrement #4 Numéro de séquence du Fichier de sortie Cette table est devenue obsolète à partir de la version V6.03 et suivantes.
RL0007	Assignment du Fichier Rapport (PPTRnn): #1 Nom du programme #2 Numéro de Fichier Cible (Rapport) MetaSuite #4 Numéro de séquence du Fichier de sortie
RL0009	Assignment du Fichier de tri Cible: #1 Nom du programme
RL0010	Assignment de fichier INI (PPTIPT): #1 Nom du programme #2 Numéro de séquence du Fichier d'entrée
RL0011	Assignment du Fichier LST (PPTLST): #1 Nom du programme #2 Numéro de séquence du Fichier de sortie
RL0012	Assignment du Fichier LOG (PPTLOG): #1 Nom du programme #2 Numéro de séquence du Fichier de sortie
RL0013	Assignment d'un Fichier d'Entrée spécifique au réseau DBMS: IDMS
RL0014	Assignment d'un Fichier d'Entrée spécifique au DBMS relationnel: DB2, SESAM
RL0015	Assignment d'un Fichier d'Entrée spécifique au DBMS hiérarchique: IMS
RL0016	Assignment d'un Fichier d'Entrée spécifique ADABAS/C:
RL0017	Exécution de programme relationnel: DB2, SESAM #1 Nom du programme #2 Numéro de version du programme
RL0018	Exécution d'un programme ADABAS/C: #1 Nom du programme #2 Numéro de version du programme
RL0019	Exécution hiérarchique d'un programme: IMS #1 Nom du programme #2 Numéro de version du programme
RL0020	Exécution d'un programme de réseau: IDMS #1 Nom du programme #2 Numéro de version du programme
RL0021	Fichier de sortie scratch pour PPTTFnn: #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite (Cette table est devenue obsolète à partir de la version V6.03 de MetaSuite et suivantes.)
RL0022	Fichier de données Cible scratch pour PPTTDnn: #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite

RL0024	Rapport de sortie scratch pour PPTRnn: #1 Nom du programme #2 Numéro de Fichier Cible (Rapport) MetaSuite
RL0025	Fichier temporaire scratch pour PPTTnn: #1 Nom du programme #2 Numéro du Fichier Source MetaSuite pour lequel un Fichier temporaire est nécessaire (dû à un Tri ou un Extrait de Fichier Source)
RL0026	Assignment du Fichier de sortie contrôlé: #1 Nom du programme #2 Numéro du Fichier Source MetaSuite contrôlé
RL0029	Retirer les Fichiers de travail de tri Cible: #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite
RL0032	Fichier de sortie scratch pour le Fichier de sortie contrôlé PPTTDnn: #1 Nom du programme #2 Numéro de Fichier Cible MetaSuite
RL0100	Création du Fichier de paramètres - 1 ^{ière} partie
RL0101	Assignment du paramètre par défaut pour les Champs de paramètres alphanumériques.
RL0102	Assignment du paramètre par défaut pour les Champs de paramètres numériques.
RL0103	Assignment du paramètre par défaut pour les Champs de paramètres de type date.
RL0199	Création du Fichier de paramètres - 2 ^{ième} partie
RL1002	Obtenir ou télécharger le Fichier Source externe #2 Numéro de séquence du Fichier d'entrée #3 Nom de Fichier local, Fichier d'entrée pour l'exécutable généré #5 Type de la Source externe Contient le mot "STD" si aucun double point n'est rencontré à la position 4 du paramètre EXTERNAL SOURCE. Si un double point est rencontré à la position 4 du paramètre EXTERNAL-SOURCE, la valeur précédant le double point est placée dans #5. #6 Fichier Source externe, nom de fichier à distance
RL1003	Assignment du Fichier Source VSAM (PPTFnn): #1 Nom du programme #2 Numéro de Fichier Source MetaSuite #3 Nom de Fichier Source MetaSuite
RL1010	Assignment du Fichier LID (PPTLID): #1 Nom du programme #2 Numéro de séquence du Fichier d'entrée
RL1012	Supprimez le Fichier Source externe téléchargé #2 Numéro de séquence du Fichier d'entrée #3 Nom de fichier local, fichier d'entrée pour l'exécutable généré - ce fichier peut être supprimé #5 Type de la Source externe Contient le mot "STD" si aucun double point n'est rencontré à la position 4 du paramètre EXTERNAL SOURCE. Si un double point est rencontré à la position 4 du paramètre EXTERNAL-SOURCE, la valeur précédant le double point est placée dans #5. #6 Fichier Source externe, nom de fichier à distance

RL9999	Fin du script: #1 Nom du programme #2 Numéro de version du programme
--------	--

16.2. Exemples

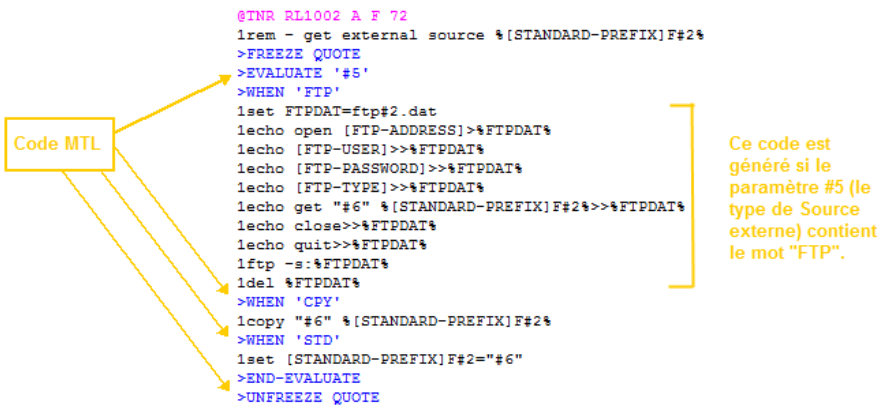
Voici quelques exemples pour vous donner une idée des possibilités.

Windows

Cet exemple utilise la fonctionnalité "EXTERNAL SOURCE".

Fichier d'entrée PPTF01	rem Windows/Fujitsu Runscript For p2569b-0001	RL0000
	set PPTIPT=NUL	RL0010
	set PPTF01=XML-XMLSAMP1	RL0003
	set PPTLOG=p2569b.log	RL0012
Fichiers de journal, de liste et de débogage	del %PPTLOG%	RL0012
	set PPTLST=p2569b.lst	RL0011
	del %PPTLST%	RL0011
	set PPTDBG=p2569b.dbg	RL0011
	del %PPTDBG%	RL0011
Fichier de sortie PPTTD01	set PPTTD01=p2569b.d01	RL0005
	del %PPTTD01%	RL0022
	rem - get external source %PPTF01%	RL1002
	set FTPDAT=ftp01.dat	RL1002
	echo open users.skynet.be>%FTPDAT%	RL1002
	echo annie.six>>%FTPDAT%	RL1002
	echo metasuite>>%FTPDAT%	RL1002
	echo binary>>%FTPDAT%	RL1002
Obtenez la Source externe via le protocole FTP	echo get "xmlsample.dat" %PPTF01%>>%FTPDAT%	RL1002
	echo close>>%FTPDAT%	RL1002
	echo quit>>%FTPDAT%	RL1002
	ftp -s:%FTPDAT%	RL1002
	del %FTPDAT%	RL1002
Paramètres de la base de données	rem - ODBC assignments	RL0001
	set @ODBC_inf=MS-ODBC.inf	RL0001
	rem - program execution	RL0001
	set @EnvSetWindow=UNUSE	RL0001
	set @MessOutFile=p2569b.err	RL0001
Exécution	p2569b.exe	RL0001
	set RC=%errorlevel%	RL0001
	rem - remove external source %PPTF01%	RL1012
Traitement Fin de Job	del %PPTF01%	RL1012
	echo %RC%	RL9999

Si nous regardons la définition de la table RL1002, nous constatons que cette table contient un mélange de paramètres spécifiques à la table (#2, #5, #6...), de paramètres MTL ([FTP-ADDRESS]...) et de paramètres de job (%PATH%, %PPTF#2%...)



UNIX / Linux

Si vous générez pour "Support Meta Suite", les noms de table sont affichés à la position 73.

# Unix/MicroFocus Runscript For c00-0001	RL0000
export dd_PPTIPT=NUL	RL0010
export dd_PPTT01=c00.t01	RL0004
export dd_PPTLOG=c00.log	RL0012
rm -f \$dd_PPTLOG	RL0012
export dd_PPTLST=c00.lst	RL0011
rm -f \$dd_PPTLST	RL0011
export dd_PPTTD01=c00.d01	RL0005
export dd_PPTR01=c00.R01	RL0007
export dd_PPTTD02=c00.d02	RL0005
rm -f \$dd_PPTTD01	RL0022
rm -f \$dd_PPTR01	RL0024
rm -f \$dd_PPTTD02	RL0022
rm -f \$dd_PPTT01	RL0025
export dd_PPTDBG=c00.dbg	RL0001
# export TMPDIR=sortwork-temp-dir	RL0001
# export COBSW=+D+S5-s-F	RL0001
# export COBEXTFHBUF=nnnnnn	RL0001
c00	RL0001
rc=\$?	RL0001
export rc	RL0001
rm -f \$dd_PPTT01	RL0025
echo "returncode is '\$rc'"	RL9999

VMS

Si vous générez pour "Support MetaSuite", les noms de table sont affichés à la position 73.

\$! VAX Runscript For ex0-0004	RL0000
\$DEL ex0.LOG	RL0000
\$DEL ex0.LST	RL0000
\$DEFINE PPT\$IPT	RL0010
\$DEFINE PPT\$F01 employee-master	RL0003
\$DEFINE PPT\$LOG ex0.log	RL0012
\$DEFINE PPT\$LST ex0.lst	RL0011
\$DEFINE PPT\$TD01 ex0.d01	RL0005
\$DEL ex0.d01	RL0022
\$RUN ex0.EXE	RL0001
\$!	RL9999

OSD/BC BS2000

```

/LOGON
/SYSFILE SYSDTA=(SYSCMD)
/OPTION DUMP=YES,MSG=FHL
/CREATE-JV JV-NAME=JV-PARM
/SET JV-LINK JV-NAME=JV-PARM
/MOD-JV JV-PARM, '..parm ..'
/SYSFILE SYSLST=EX0.LST
/ERASE EX0.LOG
/ERASE EX0.LST
/ERASE EX0.D01
/ADFL LINK=PPTF01, F-NAME=employee-master
/CRF F-NAME=EX0.D01, SUP=*PUB-DISK(VOL=*STD, -
/    SPACE=*REL (PRIM-ALLOC=32, SEC-ALLOC=16))
/ADFL LINK=PPTTD01, F-NAME=EX0.D01, BUF-LEN=*STD(SIZE=16)
/ADFL LINK=PPTIPT, F-NAME=*DUMMY
/CRF F-NAME=EX0.LOG
/ADFL LINK=PPTLOG, F-NAME=EX0.LOG
/REMARK
/ADFL LINK=COBOBJCT, F-NAME=**LOADLIB**
/ADFL LINK=BLSLIB00, F-NAME=$TSOS.SYSLNK.CRTE
/SYSFILE SYSDTA=(SYSCMD)
/START-PROGRAM FROM-FILE=-
/    *MODULE( -
/        LIBRARY=**LOADLIB** -
/        ,ELEMENT=EX0 -
/        ,PROGRAM-MODE=ANY -
/        ,RUN-MODE=ADVANCED -
/        (ALTERNATE-LIBRARIES=YES -
/        ,UNRES-EXT=DELAY -
/        ,LOAD-INFO=REFERENCE ) -
/    )
/LOGOFF

```

OS/390 Z/OS

```
//C00 JOB **ACC**, '**ID**', MSGCLASS=**MCLS**,
//      MSGLEVEL=**MLVL**, REGION=4096K,
//      CLASS=**JCLS**, NOTIFY=**USER**
//DEL   EXEC PGM=IEFBR14
//LOG   DD DSN=C00.LOG, DISP=(MOD, DELETE, DELETE),
//      SPACE=(TRK, (1, 1))
//D01   DD DSN=C00.D01, DISP=(MOD, DELETE, DELETE),
//      SPACE=(TRK, (1, 1))
//*
//D02   DD DSN=C00.D02, DISP=(MOD, DELETE, DELETE),
//      SPACE=(TRK, (1, 1))
//T01   DD DSN=C00.T01, DISP=(MOD, DELETE, DELETE),
//      SPACE=(TRK, (1, 1))
//PARM  EXEC PGM=IEBGENER, COND=(0, NE)
//SYSPRINT DD   SYSOUT=**PCLS**
//SYSIN  DD   DUMMY
//SYSUT1 DD   *
*SYS-NUMERIC-CHECK = IGNORE
*SYS-DATE-CHECK = IGNORE
/*
//SYSUT2 DD   UNIT=**TEMPDA**,
//      DISP(, PASS, DELETE),
//      DCB=(RECFM=FBA, LRECL=255, BLKSIZE=27795),
//      SPACE=(TRK, (1, 1), RLSE)
//RUN    EXEC PGM=IKJEFT01, COND=(0, NE), DYNAMNBR=20, REGION=**DB2REGN*
//STEPLIB DD DSN=**DB2LIB**, DISP=SHR
//      DD DSN=**QUAL**.USER.LOADLIB, DISP=SHR
//      DD DSN=**SRTLIB**, DISP=SHR
//      DD DSN=**CBLLIB**, DISP=SHR
//      DD DSN=**QUAL**.LOADLIB, DISP=SHR
//SYSPROC DD DSN=**DB2CLIB**, DISP=SHR
//DBRMLIB DD DSN=**QUAL**.DBRM.DB2, DISP=SHR
//SYSDBOUT DD SYSOUT=**PCLS**
//SYSPRINT DD SYSOUT=**PCLS**
//SYSOUT  DD SYSOUT=**PCLS**
//SYSTSPRT DD SYSOUT=**PCLS**
//SYSTEM  DD SYSOUT=**PCLS**
//SYSIN   DD DUMMY
//SORTLIB DD DSN=**SRTLIB**, DISP=SHR
//SORTWK01 DD UNIT=**TEMPDA**, SPACE=(**SRTTRK**, , CONTIG)
//SORTWK02 DD UNIT=**TEMPDA**, SPACE=(**SRTTRK**, , CONTIG)
//SORTWK03 DD UNIT=**TEMPDA**, SPACE=(**SRTTRK**, , CONTIG)
//SORTWK04 DD UNIT=**TEMPDA**, SPACE=(**SRTTRK**, , CONTIG)
//PPTIPT  DD DSN=*.PARM.SYSUT2,
//      DSP=SHR
//PPTLOG  DD DSN=C00.LOG, DISP=(NEW, CATLG), DCB=(RECFM=V),
//      SPACE=(TRK, (1, 1), RLSE), UNIT=SYSDA
//PPTLST  DD SYSOUT=**PCLS**
//PPTDBG  DD SYSOUT=**PCLS**
//PPTT01  DD DSN=C00.T01, DISP=(NEW, PASS),
//      SPACE=(CYL, (10, 2))
//PPTTD01 DD DSN=C00.D01, DISP=(NEW, CATLG),
//      SPACE=(TRK, (1, 1), RLSE), UNIT=SYSDA
//PPTR01  DD SYSOUT=**PCLS**
//PPTTD02 DD DSN=C00.D02, DISP=(NEW, CATLG),
//      SPACE=(TRK, (1, 1), RLSE), UNIT=SYSDA
//SYSTSIN DD *
```

```
PROFILE PREFIX(**DB2PREF**)
DSN SYS(**DB2SSID**)
BIND PLAN (C00) -
  MEMBER (C00) -
  QUALIFIER(**DB2QUAL**) -
  ACTION (REPLACE) ISOLATION (CS)
RUN PROGRAM (C00) PLAN (C00) -
  LIB ('**QUAL**.USER.LOADLIB') -
  PARM ('/')
/*
```

Modules de chaînes de caractères

Pour faciliter la manipulation des chaînes de caractères à partir d'un Programme MetaSuite, une série de sous-programmes, appelés des Modules de chaînes de caractères, peuvent être mis à la disposition du développeur. Au total, cinq modules ont été développés: un module pour les initialisations et quatre autres pour les manipulations des chaînes de caractères réelles.

17.1. Description

INITSTR	Initialise tous les Champs de paramètres. Voir INITSTR - Initialise les Champs de paramètres on page 140.
LENSTR	Définit la longueur d'une chaîne de caractères, en ignorant les espaces de fin. Voir LENSTR - Définit la longueur d'une chaîne de caractères on page 141.
SUBSTR	Renvoie une sous-chaîne de caractères à partir d'une chaîne de caractères (d'entrée) donnée. Voir SUBSTR - renvoyer une chaîne de caractères à partir d'une chaîne de caractères on page 142.
SRCHSTR	Recherche les occurrences d'un argument de recherche dans une chaîne de caractères. Voir SRCHSTR - Rechercher dans une chaîne de caractères on page 144.
REPLSTR	Remplace les occurrences d'un argument (de recherche) dans une chaîne de caractères par une chaîne de caractères de remplacement. Voir REPLSTR - remplacer une sous-chaîne de caractères dans une chaîne de caractères on page 146.

17.2. Usage

Le Fichier de Dictionnaire (MDL) suivant doit être disponible dans MetaStore:

```
DELETE FILE FunctionString
ADD FILE FunctionString TYPE FUNCTION FIXED 1038
RULE MetaSuite String Functions.
ADD RECORD String-Parms SIZE 1038
ADD FIELD String-In          POSITION 1 SIZE 256 TYPE CHARACTER
ADD FIELD String-Out         POSITION 257 SIZE 256 TYPE CHARACTER
ADD FIELD String-Search      POSITION 513 SIZE 256 TYPE CHARACTER
ADD FIELD String-Replace     POSITION 769 SIZE 256 TYPE CHARACTER
ADD FIELD String-Return      POSITION 1025 SIZE 2 TYPE BINARY
ADD FIELD String-Start       POSITION 1027 SIZE 2 TYPE BINARY
ADD FIELD String-Length      POSITION 1029 SIZE 2 TYPE BINARY
ADD FIELD String-Times       POSITION 1031 SIZE 2 TYPE BINARY
ADD FIELD String-SearchLength POSITION 1033 SIZE 2 TYPE BINARY
ADD FIELD String-ReplaceLength POSITION 1035 SIZE 2 TYPE BINARY
ADD FIELD String-Mode        POSITION 1037 SIZE 2 TYPE BINARY
```

Pour pouvoir utiliser les Fonctions de chaînes de caractères, vous devez ajouter le Fichier de Dictionnaire 'FunctionString' au programme MetaMap comme un Fichier Source normal. Le type de ce Fichier de Dictionnaire est 'FUNCTION', ce qui signifie qu'aucune donnée n'est lue de ce Fichier Source, mais que le Fichier est uniquement utilisé comme une définition d'interface pour les Modules de chaînes de caractères. Ces Champs peuvent être utilisés comme variables globales.

Assurez-vous que tous les Champs d'entrée ont une valeur avant d'appeler un des modules. Utilisez la commande "INVOKE" pour appeler le module requis, en utilisant l'Enregistrement de fonction 'String-Parms' comme paramètre.

Par exemple: INVOKE 'REPLSTR' (String-Parms)

Après cet appel, les Champs de sortie peuvent être utilisés pour un traitement ultérieur.

17.3. INITSTR - Initialise les Champs de paramètres

```
INITSTR
(
  String-In,
  String-Out,
  String-Search,
  String-Replace,
  String-Return,
  String-Start,
  String-Length,
  String-Times,
  String-SearchLength,
  String-ReplaceLength,
  String-Mode
)
```

Description

Initialise tous les Champs de paramètres.

Entrée

Ne s'applique pas

Résultats

String-In :	“
String-Out :	“
String-Search :	“
String-Replace :	“
String-Return :	0
String-Start :	1
String-Length :	-1
String-Times :	-1
String-SearchLength :	-1
String-ReplaceLength :	-1
String-Mode :	0

17.4. LENSTR - Définit la longueur d'une chaîne de caractères

```

LENSTR
(
  String-In,
  String-Out,
  String-Search,
  String-Replace,
  String-Return,
  String-Start,
  String-Length,
  String-Times,
  String-SearchLength,
  String-ReplaceLength,
  String-Mode
)

```

Description

Définit la longueur d'une chaîne de caractères d'entrée, en ignorant les espaces de fin.

Entrée

String-In	La chaîne de caractères sur la base de laquelle la longueur devrait être définie.
String-Length	Normalement, cette valeur devrait être établie à -1. Une amélioration de la performance peut être obtenue en établissant cette valeur à la longueur maximale attendue. Cependant, pour obtenir des résultats corrects, la valeur actuelle de "String-in" doit être inférieure à la valeur maximale attendue. (voir également la note 1 sous Remarques (page 142))

Résultats

String-Return	Le code retour (voir la note 2 sous Remarques (page 142)).
String-Length	La longueur de "String-In", en ignorant les espaces de fin.

Remarques

Note:

1. Ce qui se passe essentiellement c'est que la longueur moins les espaces de fin est définie pour une sous-chaîne de caractères de "String-In" commençant à la position 1 et ayant une longueur égale à la valeur d'entrée de "String-Length". Une amélioration de la performance est obtenue si la valeur d'entrée de "String-Length" peut être beaucoup plus petite que 256.
2. Si, à cause d'inconsistances dans l'entrée, LENSTR ne peut pas définir la longueur, le code de retour est établi à -2004 dans "String-Return".

Exemples

Point important: l'effet de la valeur d'entrée de String-Length

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABC abc A'						-1				
Out					0		9				
In	'ABC abc A'						15				
Out					0		9				
In	'ABC abc A'						8				
Out					0		7*				
In	'ABC abc A'						-3				
Out					-2004		-3				

* Ce résultat n'est pas correct (voir également la première note sous [Remarques](#) (page 142))

17.5. SUBSTR - renvoyer une chaîne de caractères à partir d'une chaîne de caractères

SUBSTR
(

```

String-In,
String-Out,
String-Search,
String-Replace,
String-Return,
String-Start,
String-Length,
String-Times,
String-SearchLength
String-ReplaceLength,
String-Mode
)

```

Description

Renvoie une sous-chaîne de caractères à partir d'une chaîne de caractères (d'entrée) donnée en utilisant une position de départ et une longueur.

Entrée

String-In :	La chaîne de caractères de laquelle la sous-chaîne de caractères doit être prise.
String-Start :	La position dans "String-In" à laquelle la sous-chaîne de caractères commence.
String-Length :	La longueur de la sous-chaîne de caractères à prendre. La valeur -1 indique que la sous-chaîne de caractères doit être prise de "String-In" à partir de la position "String-Start" et se terminant à la fin de "String-In", en ignorant les espaces de fin.
String-Mode :	Le mode dans lequel la sous-chaîne de caractères devrait être renvoyée. Il y a deux possibilités: 0: La sous-chaîne de caractères est renvoyée telle quelle 1: La sous-chaîne de caractères est renvoyée en majuscules. 2: La sous-chaîne de caractères est renvoyée en minuscules. Note: Les autres valeurs pour "String-Mode" sont traitées comme la valeur 0.

Résultats

String-Out :	La sous-chaîne de caractères résultante prise de "String-In".
String-Return :	Le code retour (voir la note 1 sous Remarques (page 143)).

Remarques

Note:

1. Si, à cause d'inconsistances dans l'entrée, SUBSTR ne peut pas renvoyer une sous-chaîne de caractères, le code de retour est établi à **-2001** dans "String-Return".

Exemples

Point important: L'effet de String-Mode et String-Length

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABCabcA'					3	3				0
Out		'Cab'			0						
In	'ABCabcA'					3	3				1
Out		'CAB'			0						
In	'ABCabcA'					3	3				2
Out		'cab'			0						
In	'ABCabcA'					3	-1				0
Out		'CabcA'			0						

17.6. SRCHSTR - Rechercher dans une chaîne de caractères

SRCHSTR

```
(
  String-In,
  String-Out,
  String-Search,
  String-Replace,
  String-Return,
  String-Start,
  String-Length,
  String-Times,
  String-SearchLength,
  String-ReplaceLength,
  String-Mode
)
```

Description

Recherche des occurrences d'un argument de recherche "String-Search" dans une chaîne de caractères d'entrée "String-In". L'opération commence à "String-Start" et continue sur une longueur égale à la valeur de "String-Length". Le nombre maximal d'occurrences de "String-Search" à rechercher est défini par la valeur d'entrée de "String-Times".

Entrée

String-In : La chaîne de caractères dans laquelle s'effectuera la recherche.

String-Search : L'argument de recherche.

String-Start : La position dans "String-In" à laquelle commencera la recherche.

String-Length : La longueur sur laquelle la recherche sera effectuée.
La valeur -1 indique que la recherche doit être exécutée jusqu'à la fin de "String-In", en ignorant les espaces de fin.

- String-Times :** Le nombre d'occurrences de "String-Search" à rechercher.
La valeur -1 indique que la recherche doit être exécutée pour toutes les occurrences.
- String-SearchLength :** La longueur de l'argument de recherche. La valeur -1 indique que la recherche doit être exécutée sur l'argument de recherche sans les espaces de fin.
- String-Mode :** Le mode dans lequel la recherche doit être exécutée.
Il y a deux possibilités:
0: La recherche tient compte de la casse.
1: La recherche ne tient pas compte de la casse.
Note: Les autres valeurs pour "String-Mode" sont traitées en tenant compte de la casse (valeur 0).

Résultats

- String-Return :** Le code retour (voir la note 3 sous [Remarques](#) (page 145)).
- String-Start :** La position de départ dans "String-In" de la dernière occurrence trouvée de "String-Search". "String-Start" contiendra la valeur 0 si aucune occurrence de "String-Search" n'est trouvée.
- String-Times :** Le nombre d'occurrences de "String-Search" trouvées. Ce nombre ne peut pas dépasser le nombre d'occurrences recherché.

Remarques

Note:

1. Les espaces de fin peuvent être inclus dans un argument de recherche en spécifiant une "String-SearchLength" supérieure à la longueur actuelle de "String-Search", c.-à-d. la longueur moins les espaces de fin.
2. Si "String-SearchLength" est inférieure à la longueur actuelle de "String-Search", seule la première partie de "String-SearchLength" de "String-Search" sera utilisée pour exécuter la recherche.
3. Si, à cause d'inconsistances dans l'entrée, SRCHSTR ne peut pas exécuter une recherche, le code de retour est établi à -2003 dans "String-Return".

Exemples

- Point important: L'effet de String-Mode

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABCabcA'		'BC'			1	7	-1	2		0
Out					0	2		1			
In	'ABCabcA'		'BC'			1	7	-1	2		1
Out					0	5		2			

- Point important: L'effet de String-Times

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABAACAD'		'A'			1	7	-1	1		0
Out					0	6		4			
In	'ABAACAD'		'A'			1	7	3	1		0
Out					0	4		3			

- Point important: L'effet de String-SearchLength et de String-Length

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABA ACA '		'A'			1	8	-1	2		0
Out					0	7		2			
In	'ABA ACA '		'A'			1	8	-1	1		0
Out					0	7		4			
In	'ABA ACA '		'A'			1	8	-1	0*		0
Out					-2003	0		0			
In	'ABA ACA '		'A'			1	-1	-1	2		0
Out					0	3		1			

* 'String-SearchLength' ne peut pas être 0

17.7. REPLSTR - remplacer une sous-chaîne de caractères dans une chaîne de caractères

REPLSTR

```
(
  String-In,
  String-Out,
  String-Search,
  String-Replace,
  String-Return,
  String-Start,
  String-Length,
  String-Times,
  String-SearchLength,
  String-ReplaceLength,
  String-Mode
)
```

Description

Remplace l'argument de recherche "String-Search" dans une chaîne de caractères d'entrée "String-In" par "String-Replace". L'opération commence à "String-Start" et continue sur une longueur égale à la valeur de "String-Length". Le nombre maximal d'occurrences de "String-Search" à remplacer est défini par la valeur d'entrée de "String-Times". Le résultat de l'opération est renvoyé dans "String-Out".

Les longueurs de "String-Search" et "String-Replace" peuvent être différentes. Dans ce cas, si une occurrence de "String-Search" est trouvée, elle sera remplacée par "String-Replace" et la partie restante de "String-In" sera décalée vers la gauche ou vers la droite.

Il est également possible de spécifier une chaîne de caractères vide pour "String-Replace" en établissant la valeur de "String-ReplaceLength" à 0. Cela retirera "String-Search" de "String-In".

Entrée

String-In	La chaîne de caractères dans laquelle s'effectuera la recherche et le remplacement.
String-Search	L'argument de recherche.
String-Replace	La chaîne de caractères de remplacement, c.-à-d. la chaîne de caractères qui remplacera l'argument de recherche.
String-Start	La position dans "String-In" à laquelle commencera la recherche et le remplacement.
String-Length	La longueur sur laquelle s'effectuera la recherche et le remplacement. La valeur -1 indique que la recherche et le remplacement doivent être exécutés jusqu'à la fin de "String-In", en ignorant les espaces de fin.
String-Times	Le nombre d'occurrences de "String-Search" à remplacer. La valeur -1 indique que la recherche et le remplacement doivent être exécutés pour toutes les occurrences.
String-SearchLength	La longueur de l'argument de recherche. La valeur -1 indique que la recherche doit être exécutée sur l'argument de recherche sans les espaces de fin.
String-ReplaceLength	La longueur de la chaîne de caractères de remplacement. La valeur -1 indique que l'argument de recherche doit être remplacé par la chaîne de caractères de remplacement moins les espaces de fin.
String-Mode	Le mode dans lequel la recherche doit être exécutée. Il y a deux possibilités: 0: La recherche tient compte de la casse. 1: La recherche ne tient pas compte de la casse. Note: Les autres valeurs pour "String-Mode" sont traitées en tenant compte de la casse (valeur 0).

Résultats

String-Out	La chaîne de caractères résultante après l'exécution de la recherche et du remplacement.
String-Return	Le code de retour (voir également les notes 5 et 6 sous Remarques (page 145)).
String-Times	Le nombre d'occurrences de "String-Search" trouvées et remplacées. Ce nombre ne peut pas dépasser le nombre d'occurrences recherché.

Remarques

Note:

1. Les espaces de fin peuvent être inclus dans un argument de recherche en spécifiant une "String-SearchLength" supérieure à la longueur actuelle de "String-Search", c.-à-d. la longueur moins les espaces de fin.
2. Les espaces de fin peuvent être inclus dans la chaîne de caractères de remplacement en spécifiant une "String-ReplaceLength" supérieure à la longueur actuelle de "String-Replace", c.-à-d. la longueur moins les espaces de fin.

3. Si "String-SearchLength" est inférieure à la longueur actuelle de "String-Search", seule la première partie de "String-SearchLength" de "String-Search" sera utilisée pour exécuter la recherche.
4. Si la "String-ReplaceLength" est inférieure à la longueur actuelle de "String-Replace", seule la première partie de "String-ReplaceLength" de "String-Replace" sera utilisée pour exécuter la recherche.
5. Si, à cause d'inconsistances dans l'entrée, REPLSTR ne peut pas exécuter une **recherche**, le code de retour est établi à -2003 dans "String-Return".
6. Si, à cause d'inconsistances dans l'entrée, REPLSTR ne peut pas exécuter un **remplacement**, le code de retour est établi à -2002 dans "String-Return".

Exemples

- Point important: L'effet de String-Mode

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABCabcA'		'BC'	'Xy'		1	7	-1	2	2	0
Out		'AXyabcA'			0			1			
In	'ABCabcA'		'BC'	'Xy'		1	7	-1	2	2	1
Out		'AXyaXyA'			0			2			

- Point important: L'effet de String-Times

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABAACAD'		'A'	'x'		1	7	-1	1	1	0
Out		'xBxxCxD'			0			4			
In	'ABAACAD'		'A'	'x'		1	7	3	1	1	0
Out		'xBxxCAD'			0			3			

- Point important: L'effet de String-ReplaceLength

	String-In	String-Out	Str-Srch.	Str-Repl.	Str-RC	Str-Start	Str-Lngth	Str-Tms	Str-Srch. Lngth	Str-Repl. Lngth	Str-Mode
In	'ABCabcA'		'a'	'Xy'		1	7	-1	1	3	0
Out		'ABCXy bcA'			0			1			
In	'ABCabcA'		'a'	'Xy'		1	7	-1	1	2	0
Out		'ABCXybcA'			0			1			
In	'ABCabcA'		'a'	'Xy'		1	7	-1	1	1	0
Out		'ABCXbcA'			0			1			
In	'ABCabcA'		'a'	'Xy'*'		1	7	-1	1	0	0
Out		'ABCbcA'			0			1			

Toute autre valeur pour "String-Replace" aura le même résultat puisque "String-ReplaceLength" est établi à 0.

Appeler le Générateur en mode batch

Le Générateur est appelé en batch en exécutant le programme *MSBGEN.EXE* à partir de l'invite DOS. Ce programme est disponible dans le dossier d'Installation MetaSuite.

Il permet de:

- créer et maintenir le Dictionnaire du Générateur
- régénérer des fichiers MXL.

18.1. Préparation

Avant d'exécuter le programme *MSBGEN.EXE*, copiez le fichier de commandes, p.ex. un fichier MIL ou MXL, vers le dossier temporaire *TMP* sous *<Installation>/GEN**. Ensuite, renommez le fichier *PPTOUT.TMP*. *PPTOUT.TMP* est le fichier de sortie standard de *MetaMap* et sera utilisé comme fichier d'entrée pour *MSBGEN.EXE*.

18.2. Utiliser MSBGEN.EXE

Après avoir exécuté *MSBGEN.EXE*, les informations de sortie du processus de génération se trouvent dans *PPTIN.TMP*. *PPTIN.TMP* est le fichier d'entrée pour *MetaMap* et le fichier de sortie pour *MSBGEN.EXE*.

Ouvrez une session MS-DOS et accédez au dossier d'installation MetaSuite.

Exécutez le programme en utilisant le format de commande suivant:

MSBGEN TYPE-GÉNÉRATION MODE-INFORMATION INFORMATION-RÉPERTOIRE

La table suivante fournit un aperçu des types de génération disponibles.

Type	Description
MIL	Script d'installation MetaSuite, écrit en langage MIL
MXL	Script d'exportation MetaSuite, écrit en langage de définition MetaSuite et en langage de spécification MetaSuite.
MDL	Langage de définition MetaSuite, seulement à des fins de test.
MSL	Langage de spécification MetaSuite , seulement à des fins de test.

La table suivante fournit un aperçu des modes d'information disponibles.

Mode	Description
NOTRD	Mode Standard, pas de mode Trace, aucune information de débogage
TRACE	Mode Trace.
DEBUG	Information de débogage pour vérifier quelles tables sont générées par le Générateur.
TRDBG	Une combinaison du mode Trace et du mode Débogage

La table suivante donne un aperçu des répertoires possibles.

Répertoire	Description
Temporaire	Dossier temporaire utilisé pour les Fichiers temporaires. Le Fichier <i>mscopy.mil</i> sera également placé ici, si la commande MIL "Copy Table" a été utilisée.
Répertoire MGL	Ce répertoire contiendra les fichiers MGL, si vous générez un fichier MXL ou MSL.
Répertoire MRL	Ce répertoire contiendra les fichiers MRL, si vous générez un fichier MXL ou MSL.
Répertoire DCT	Ce répertoire est utilisé par MetaSuite pour récupérer ou stocker le fichier du Dictionnaire du Générateur (<i>msdict.dct</i>).
Répertoire DDL	Dans ce répertoire, MetaSuite trouvera les fichiers Source pour alimenter le fichier du Dictionnaire du Générateur (<i>msini.ddl</i> , <i>msmgl.ddletmsmrl.ddl</i>). Ce répertoire doit être spécifié si la commande MIL "NEW" est utilisée.
Répertoire d'installation	Nom du répertoire (nom abrégé) dans lequel MetaSuite est installé. Ce répertoire doit être spécifié si la commande MIL "NEW" est utilisée.

18.3. Exemple

1. Copiez un Fichier MXL vers <Installation>\GEN*\TMP.

2. Renommez le fichier en pptout.tmp.

3. Saisissez la commande suivante à l'invite DOS:

```
MSBGEN MXL DEBUG '<DOC>\GEN*\TMP' '<DOC>\GEN*\MGL' '<DOC>\GEN*\MRL' '<DOC>\GEN*\DCT'
'<INSTALLATION>\GEN*\DLL' '<INSTALLATION>'
```

Par exemple:

```
SET MSINS=D:\PROGRAM FILES\IKAN SOLUTIONS\METASUITE 813
```

```
CD /D "%HOMEDRIVE%\HOMEPATH%\DOCUMENTS\METASUITE"
```

```
COPY MXL\%1.MXL .\GENIBM_WINDOWS\TMP\PPTOUT.TMP
```

```
"%MSINS%\MSBGEN.EXE" MXL NOTRD ^
```

```
".\GENIBM_WINDOWS\TMP" ^
```

```
".\GENIBM_WINDOWS\MGL" ^
```

```
".\GENIBM_WINDOWS\MRL" ^
```

```

".\GENIBM_WINDOWS\DCI" ^
"%MSINS%\GENIBM_WINDOWS\DDL" ^
"%MSINS%"
ECHO %ERRORLEVEL%

```

4. Accédez au dossier <Doc>\GEN*\TMP
Utilisez un éditeur de texte pour vérifier le fichier *pptin.tmp* (p.ex. Notepad).
5. Accédez au dossier <Doc>\GEN*\MGL
Vérifiez le Fichier *EX0.MGL* pour voir la Source COBOL générée.
6. Accédez au dossier <Doc>\GEN*\MRL
Vérifiez le fichier *EX0.MRL* pour afficher le script d'exécution généré.

Messages du Générateur

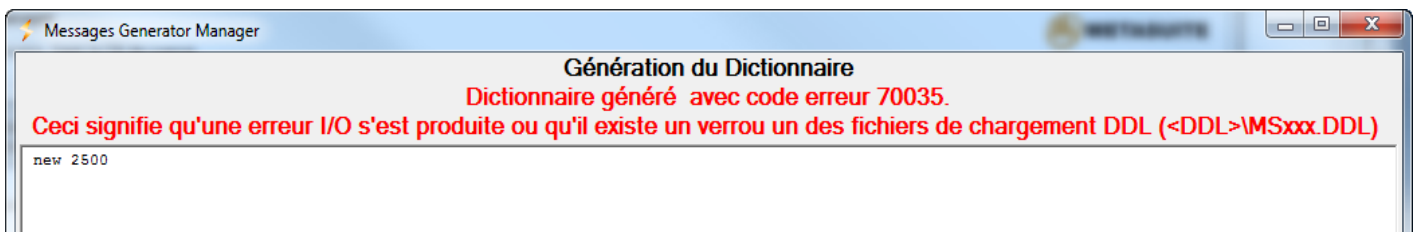
Cette section décrit:

- [Codes de retour du Générateur](#) (page 152)
- [Messages d'erreurs de syntaxe du Générateur](#) (page 155)
- [Messages d'erreur du Dictionnaire](#) (page 156)
- [La liste des messages du Générateur](#) (page 157)

19.1. Codes de retour du Générateur

Les Codes de retour du Générateur identifient des problèmes potentiels dans les Fichiers système du Générateur au moment de la génération.

Les Codes de retour du Générateur sont affichés dans le format suivant:



Contactez le service de support technique de MetaSuite et conservez une copie du Fichier causant le problème à portée de main pour complément d'enquête.

Les Codes de retour du Générateur sont toujours composés de 5 chiffres. Le chiffre de début se réfère au(x) Fichier(s) concerné(s), comme indiqué dans le tableau suivant:

Premier chiffre	Fichier(s) concerné(s)
1	Liste de sortie du Générateur TEMPDIR\PPTIN.TMP
2	Fichier de chargement du Dictionnaire au moment d'installation: <Installation>\...\DCT\MSDICT.DCT
3	Fichier de chargement du Dictionnaire au moment de la génération: <Installation>\...\DCT\MSDICT.DCT
4	Fichier de commandes (Fichier de type MXL ou MIL): TEMPDIR\PPTOUT.TMP
5	un des fichiers générés, MGL ou MRL: <Installation>\...\MGL\mxlname.MGL <Installation>\...\MRL\mxlname.MRL
6	Fichier de sauvegarde TEMPDIR\mscopy.mil
7	Un des Fichiers de chargement DDL, INI, MGL ou MRL: <Installation>\...\DDL\MSINI.DDL <Installation>\...\DDL\MSMGL.DDL <Installation>\...\DDL\MSMRL.DDL
8	Le dossier "dump" au moment de la génération. Ce dossier est utilisé pour placer l'image mémoire demandée, lorsque la commande MTL ">DUMP" est exécutée. Défaut: TEMPDIR

Note: Dans les références ci-dessus, **TEMPDIR** pointe vers le répertoire défini dans la variable "TEMP system/user".

Les quatre chiffres suivants correspondent à un des codes suivants:

Code à quatre chiffres	Description
0002	Code de retour ne s'appliquant qu'aux Fichiers indexés. Signification: • Pour une instruction READ, la valeur de la clé courante est égale à la valeur de cette même clé dans l'Enregistrement suivant dans la clé de référence courante. • Pour une instruction WRITE ou REWRITE, l'Enregistrement qui vient d'être écrit a créé une clé en double pour au moins une Clé d'enregistrement alternative pour laquelle les valeurs en double sont autorisées.
0004	La longueur de l'Enregistrement en cours de traitement n'est pas conforme aux attributs de Fichier fixe de ce Fichier.
0005	Le Fichier optionnel référencé n'est pas présent au moment où l'instruction OPEN est exécutée.
0007	Uniquement pour des Fichiers séquentiels. Pour une instruction OPEN ou CLOSE avec la phrase REEL/UNIT du Fichier référencé est un support "NON-REEL/UNIT".
0010	Aucun Enregistrement logique suivant n'est présent. La fin du Fichier est atteinte.

Code à quatre chiffres	Description
0014	Uniquement pour des Fichiers relatifs. Le nombre de chiffres significatifs dans le nombre de l'Enregistrement relatif est plus grand que la taille de l'élément de données de clé relative décrit pour ce Fichier.
0021	Uniquement pour des Fichiers accédés de manière séquentielle: indique une erreur de séquence. Les exigences de clé ascendante pour les valeurs de Clé d'enregistrement successives ont été violées, ou, la valeur de la Clé d'enregistrement principale a été modifiée par un programme COBOL entre l'exécution réussie d'une instruction READ et l'exécution de l'instruction REWRITE suivante pour ce Fichier.
0022	Uniquement pour les Fichiers indexés et relatifs. Indique la présence d'une clé en double. Une tentative a été faite pour stocker un Enregistrement qui créerait soit une clé en double dans le Fichier indexé ou le Fichier relatif soit une Clé d'enregistrement alternative en double qui n'autorise pas les valeurs en double.
0023	Indique qu'aucun Enregistrement n'est trouvé. Une tentative a été faite pour accéder à un Enregistrement, identifié par une clé, et cet Enregistrement n'existe pas dans le Fichier. Alternativement, une tentative d'exécuter des opérations START et READ a été faite sur un Fichier d'entrée optionnel qui n'est pas présent.
0024	Uniquement pour les Fichiers relatifs et indexés: Indique un dépassement des limites résultant d'une des conditions suivantes: • Une tentative a été faite pour écrire au-delà des limites définies à l'extérieur d'un Fichier. • Une opération WRITE séquentielle a été tentée sur un Fichier relatif, mais le nombre de chiffres significatifs dans le numéro d'Enregistrement relatif est plus grand que la taille de l'élément de données de clé relative décrit pour le Fichier.
0030	L'exécution de l'instruction E/S n'a pas réussi à cause d'un dépassement des limites pour un Fichier Séquentiel ou à cause d'une erreur E/S, telle qu'une erreur de parité de vérification des données ou une erreur de transmission.
0034	L'opération E/S a échoué à cause d'une violation des limites. Cette condition indique qu'une tentative a été faite pour écrire au-delà des limites définies à l'extérieur d'un Fichier.
0035	Une opération OPEN avec les phrases I-O, INPUT ou EXTEND a été tentée sur un Fichier non-OPTIONAL qui n'est pas présent.
0037	Une opération OPEN a été tentée sur un Fichier qui ne supporte pas le mode d'ouverture spécifié dans l'instruction OPEN.
0038	Une opération OPEN a été tentée sur un Fichier fermé auparavant par un verrou.
0039	Un conflit a été détecté entre les attributs du Fichier fixe et les attributs spécifiés pour ce Fichier dans le programme.
0041	Une opération OPEN a été tentée sur un Fichier déjà ouvert.
0042	Une opération CLOSE a été tentée sur un Fichier déjà fermé.
0043	Pour les Fichiers en mode d'accès séquentiel: la dernière instruction E/S exécutée pour le Fichier, avant l'exécution d'une instruction DELETE ou REWRITE, n'était pas une instruction READ.
0044	Il y a une violation des limites.
0046	Une opération READ séquentielle a été tentée sur un Fichier ouvert dans le mode INPUT ou le mode I-O, mais aucun Enregistrement suivant valide n'est défini.

Code à quatre chiffres	Description
0047	Une opération READ ou START a été tentée sur un Fichier n'étant pas ouvert en mode INPUT ou I-O.
0048	Une opération WRITE a été tentée sur un Fichier n'étant pas ouvert en mode OUTPUT, I-O ou EXTEND, ou, sur un Fichier ouvert en mode I-O en mode d'accès séquentiel.
0049	Une opération DELETE ou REWRITE a été tentée sur un Fichier qui n'est pas ouvert en mode I-O.

Si le Code retour est égal à "-1", une erreur de sous-système s'est produite au moment de la génération.

19.2. Messages d'erreurs de syntaxe du Générateur

Cette section liste tous les messages signalant les problèmes concernant les commandes du Générateur au moment de la génération. Si le Générateur MetaSuite est démarré à partir de MetaMap, les messages s'afficheront dans la partie du Générateur MetaSuite de la fenêtre des messages dans MetaMap.

Les erreurs de syntaxe du Générateur sont très rares si le Modèle MetaSuite est développé en utilisant MetaMap.

Le message d'erreur actuel peut être précédé d'un message dans le format suivant:

SYNTAX ERROR m OCCURED NEAR : xxx

Où:

- m = numéro d'erreur séquentiel
- xxx = la chaîne de caractères où l'erreur s'est produite

Exemple:

SYNTAX ERROR 1 OCCURED NEAR : XIZE

Le message d'erreur même a le format suivant:

PgmNnnL m Message_Text

où:

- Pgm = se réfère au module PPTPgm.DLL qui traite la commande
Note: Ces 3 premiers caractères sont des indicateurs utiles pour le service de support technique.
- Nnn : Numéros de messages généraux
- L : Le niveau d'erreur
- m : le compte d'erreurs pour l'instruction MXL courante
- Message_Text : description de l'erreur

Exemple:

FES109E 1 KEYWORD EXPECTED

Les niveaux d'erreur suivants existent:

Niveau d'erreur	Description
I (informatif)	Un message informatif fournit des informations utiles.

Niveau d'erreur	Description
W (avertissement)	Un avertissement indique une incohérence dans la commande. Le système exécutera une action corrective par défaut (habituellement indiquée par un message) et la commande sera traitée. Vous devez vérifier tous les avertissements pour vous assurer que la commande a été exécutée telle que prévue.
E (erreur grave)	Une erreur grave est une erreur qui empêche le système de traiter la commande. Si la première erreur grave est rencontrée, la génération du programme est arrêtée, mais la vérification de la syntaxe continue pour le reste de la codification de votre programme. Vous devez à nouveau codifier et saisir la commande erronée.
F (Erreur irrécupérable)	Une erreur irrécupérable indique soit que la fonctionnalité de sécurité du système n'a pas pu identifier votre Utilisateur, soit qu'il y a un problème sérieux avec votre Dictionnaire. Dans les deux cas, l'exécution sera arrêtée immédiatement, sans traitement additionnel des commandes. Dans le premier cas, vous devez fournir une identification appropriée ou vous devez contacter votre Administrateur Système pour établir un mot de passe et un ID Utilisateur valides.

19.3. Messages d'erreur du Dictionnaire

Les messages d'erreur du Dictionnaire sont des erreurs causées par un Dictionnaire erroné ou inconsistant. Ces erreurs ne sont pas numérotées parce que les messages numérotés font eux-mêmes partie du Dictionnaire. Si le Dictionnaire est erroné, les messages numérotés ne sont pas accessibles.

Le tableau suivant donne un aperçu des Messages d'erreur du Dictionnaire:

Message(s)	Signification
F DICTIONARY I/O ERROR ON KEY 1 1 0 F FAILED I/O FUNCTION READ F RUN ABORTED	Il existe une incohérence dans votre Dictionnaire. Demandez à votre Administrateur MetaSuite de reconstruire le Dictionnaire.
F DICTIONARY UPDATE FLAG FOUND ALREADY SET WHEN DICTIONARY WAS OPENED. EITHER CURRENTLY IN USE, OR A PREVIOUS UPDATE JOB HAS BEEN INTERRUPTED.	Le Dictionnaire est supprimé ou corrompu. Demandez à votre Administrateur Système de restaurer le Dictionnaire.
F DICTIONARY INCOMPATIBLE WITH : Version du Générateur F PLEASE UPGRADE DICTIONARY TO : Version du Générateur	Le Dictionnaire n'est pas compatible avec le Générateur parce qu'il n'est pas mis à jour. Mettez à jour le Dictionnaire à un autre numéro de construction ou à un autre numéro de version. La version et le numéro de construction minimaux sont indiqués dans ce message.
F DICTIONARY INVALID	Le Dictionnaire n'est pas compatible avec le Générateur, mais la cause ne peut pas être déterminée. Créer un nouveau Dictionnaire.

19.4. La liste des messages du Générateur

Cette section contient la liste complète des messages du Générateur. Pour plus de facilité, ils ont été regroupés en sections, chacune contenant 50 messages.

- [Messages 001 à 050](#) (page 157)
- [Messages 051 à 100](#) (page 160)
- [Messages 101 à 150](#) (page 163)
- [Messages 151 à 200](#) (page 167)
- [Messages 201 à 250](#) (page 169)
- [Messages 251 à 300](#) (page 172)
- [Messages 301 à 350](#) (page 174)
- [Messages 351 à 400](#) (page 177)
- [Messages 401 à 450](#) (page 179)
- [Messages 451 à 500](#) (page 182)
- [Messages 501 à 550](#) (page 184)
- [Messages 551 à 600](#) (page 186)
- [Messages 601 à 650](#) (page 189)
- [Messages 651 à 700](#) (page 193)
- [Messages 701 à 711](#) (page 199)

Messages 001 à 050

Nombre	Message	Explication
001	La Date n'est pas en format Y2K	Ce message d'erreur est retourné pour chaque Champ de type date non-Y2K utilisé dans le programme.
002	Le nom commence avec un caractère non valide	Le nom de l'objet n'est pas conforme aux normes de nommage: le premier caractère est erroné.
003	Le nom contient un caractère non valide	Le nom de l'objet n'est pas conforme aux normes de nommage: un des caractères est erroné.
004	Nom trop long	
005	Littéral doit être entre parenthèses.	
006	Le nom ne peut pas se terminer par un trait d'union	Le nom d'un objet se termine en trait d'union, ce qui est interdit.
007	La valeur numérique contient un caractère non valide	
008	Trop de points décimaux	Une valeur décimale contenant plus d'un point décimal (.) est fournie.
009	L'opérateur n'est pas suivi d'un espace	Ce message est retourné si un opérateur n'est pas suivi d'un espace. Vous devez toujours ajouter un espace après un opérateur (qui sera = + - * / **).

Nombre	Message	Explication
010	Le nom est déjà défini dans le Dictionnaire	Le nom de l'objet à ajouter existe déjà dans le Dictionnaire technique. Chaque objet dans le Dictionnaire technique doit avoir un nom unique. Vous pouvez renommer l'objet vous-même ou vous pouvez saisir au début du Fichier de maintenance la commande 'DUPLICATES'. Dans ce dernier cas, le Générateur MetaSuite générera un nouveau nom pour l'objet en double en ajoutant un suffixe.
011	Erreur E/S de dictionnaire	Ce message est retourné si un objet n'est pas défini dans le Dictionnaire. Ce message est souvent accompagné du message numéro 35. Veuillez vérifier le nom de l'objet saisi, ou créez une liste de tous les objets existants dans le Dictionnaire technique pour savoir quels objets y sont définis.
012	#1 non défini dans le Dictionnaire	Le Dictionnaire du Générateur msdict.dct ne peut pas être accédé pour des raisons inconnues. Vérifiez les droits d'accès (droits d'écriture), la disponibilité du Fichier, la connexion au réseau etc. Si aucune raison externe n'est trouvée, il est possible que le Dictionnaire soit plein ou corrompu. Dans ce cas: Veuillez retirer le Fichier et construire un nouveau Dictionnaire.
013	Séquence de commandes non valide	Un programme MetaSuite est structuré suivant une séquence spécifique composée de 4 parties. Ce message est retourné si la séquence des commandes dans le MSL n'est pas respectée.
014	Fin de commande ou REMARKS attendu	Ce message est retourné si la commande contient trop d'arguments.
015	Mot clé en double	Ce message est retourné si un mot clé est utilisé deux fois (ou plus). Vous devez retirer une des occurrences du mot clé (y compris ses caractéristiques).
016	Nom de fichier attendu	L'instruction ADD FILE ne contient pas de nom, ou le nom spécifié n'est pas valide.
017	Nom attendu	Ce message général est retourné si aucun nom n'est spécifié pour un objet ou si le nom spécifié n'est pas valide. Vous devez construire les commandes de maintenance en utilisant MetaStore Manager et les commandes de programme en utilisant MetaMap.
018	Commande incomplète	Ce message général est retourné si certains mots clé obligatoires ne sont pas fournis dans la commande. Vous devez construire les commandes de maintenance en utilisant MetaStore Manager et les commandes de programme en utilisant MetaMap.
019	Usage de mot clé non valide	
020	Argument numérique attendu	Ce message est retourné si un mot clé attend une valeur numérique.
021	Nom de champ attendu	Ce message est accompagné du message <i>SYNTAX ERROR n OCCURED NEAR : xxx</i> . Il est attendu que xxx doit être un nom de Champ, ce qui n'est pas le cas.
022	Clé de fichier non permise	

Nombre	Message	Explication
023	Type de fichier non valide (ou manquant)	Ce message est retourné si le TYPE d'un FILE (Fichier) ne fait pas partie des types de fichier connus, ou si le type de Fichier est absent.
024	L'usage de HALT n'est pas permis pour les structures ou les fichiers "FUNCTION"	L'instruction HALT ne peut pas être appliquée sur un Fichier qui n'est pas ouvert ou lu. Arrêter (HALT) un Fichier signifie que le processus de lecture du Fichier est arrêté. Par conséquent: si le Fichier n'est pas lu, une instruction HALT ne peut pas être exécutée.
025	Limite maximale de la taille d'Enregistrement atteinte	
026	Position requise par rapport à	
027	Mode non valide ou non supporté	Ce message est retourné si le MODE d'un Fichier n'est pas EBCDIC ou ASCII, ou si le MODE spécifié n'est pas valide pour le Générateur MetaSuite sélectionné.
028	VALUE doit être plus grande que zéro	La valeur assignée à certaines caractéristiques ne peut pas être "0". Modifiez la valeur en une valeur valide.
029	Doit être une valeur entière	
030	Option de liste non valide	
031	Valeur de Clé d'enregistrement en conflit avec le type	
032	Le Nom de la table doit comporter 6 caractères	Une table contenant un modèle de codification pour le Dictionnaire technique (les commandes d'installation) doit avoir un nom d'exactly 6 caractères.
033	Parenthèse de gauche attendue	
034	En conflit avec une saisie antérieure	
035	Aucune définition valide demandée	Ce message est retourné s'il n'y a pas d'objets en résultat pour une liste d'objets.
036	Trop de définitions demandées	
037	La valeur numérique contient trop de chiffres	
038	Fraction manquante	
039	Littéral mal délimité	Ce message est retourné si une constante alphanumérique n'est pas correctement placée entre guillemets simples.
040	Usage de masque de date en conflit avec	
041	Non permis après SHORT	Erreurs de syntaxe dans la commande FIELDS dans le Rapport.
042	La valeur de DBNAME est trop longue	La valeur de DBNAME est dépassée. Le nombre de caractères permis dépend du type de base de données.

Nombre	Message	Explication
043	HALT <target> est remplacé automatiquement par HALT ALL	Un Fichier Cible qui est arrêté ne cause pas l'arrêt des Fichiers Source. Les Enregistrements de Fichier Source ignorés seront traités comme des Enregistrements "exclus". Donc, dans le cas d'une instruction HALT <FichierCible x>, tous les Fichiers Source seront exécutés en boucle jusqu'à ce qu'ils atteignent la fin (la condition EOF). Si tous les Fichiers Source sont des Fichiers de fonction et des Fichiers d'Écriture spéciale uniquement (qui ne sont pas lus), des Fichiers de paramètre et des Matrices (qui sont lus et fermés avant le traitement principal), cette condition EOF ne sera jamais établie à vraie et le programme se poursuivra dans une boucle infinie. Pour éviter cette situation, l'instruction HALT <FichierCible x> est modifiée en HALT ALL.
044	Mot clé ou nom attendu	
045	Virgule attendue	
046	EXCLUDE non permis	
047	Littéral ou nom attendu	Ce message est retourné si le littéral ou le nom ne contient pas de valeur correcte.
048	La taille du Champ clé par défaut est dépassé de	Il y a un problème avec la longueur du Champ clé.
049	Entrées restantes ignorées	Si une erreur précédente s'est produite, les mots clé sur la commande sont ignorés. Vous devez d'abord corriger l'erreur précédente avant de pouvoir continuer.
050	---	

Messages 051 à 100

Nombre	Message	Explication
051	Virgule ou parenthèse de droite attendue	
052	Clé de licence incorrecte	Ce message est retourné si vous n'avez pas ajouté une clé de licence, ou si la clé de licence ajoutée n'est pas valide. La licence courante est périmée.
053	Les positions décimales sont en conflit avec	Seuls les types de données numériques peuvent utiliser l'option DECIMAL. Ce message est retourné si l'option DECIMAL est utilisée dans une définition de Champ pour laquelle elle ne peut pas être utilisée.

Nombre	Message	Explication
054	Parenthèse de droite attendue	Ce message est retourné si des parenthèses de droite sont attendues mais qu'elles ne sont pas présentes. Une parenthèse de droite est utilisée à la fin d'un ensemble logique de commandes MetaSuite, telle qu'une ligne de détail ou une ligne de total dans une Fichier Cible. En général, cette erreur est liée à une erreur en amont qui oblige le Générateur MetaSuite à ignorer les parenthèses de droite qui sont présentes. Vous devez d'abord corriger l'erreur précédente.
055	Un seul caractère permis	
056	#1 non permis dans le bloc FOR END-FOR requis.	Cette instruction n'est pas permise avant la fermeture du bloc FOR.
057	NEXT est une instruction obsolète	Il est conseillé de structurer cette procédure au lieu d'utiliser une expression plus ancienne telle que NEXT.
058	Objet de transformation:	Ce message est utilisé pour la sortie des numéros de transformation si la génération est exécutée en mode TRACE.
059	La table des alias est pleine	Les noms d'Enregistrement, auxquels sont ajoutés le signe "#" et un numéro de Champ (généré par MetaMap), sont remplacés par un Nom d'alias en interne. Le nombre de Noms d'alias est limité à 500.
060	Cette fonctionnalité n'est plus supportée.	
061	Le fichier contenant ce champ n'est pas utilisé	
062	Symbole d'assignation attendu	Le signe d'égalité "=" suivi d'un espace est considéré comme un symbole d'assignation. Si le signe "=" est présent uniquement et que la valeur à assigner est concaténée au signe "=", le signe "=" ne sera pas considéré comme un symbole d'assignation.
063	Taille d'Enregistrement non valide	La taille de l'Enregistrement est égale à zéro, négative ou plus grande que la taille du Fichier.
064	Clause CODE non compatible avec #1	Si un numéro a reçu l'attribut "CODE", les autres spécifications telles que "NATIVE", "TIMESTAMP", etc. écraseront la spécification "CODE".
065	L'instruction HALT est requise pour éviter une boucle infinie	Si tous les Fichiers Source sont des Fichiers de fonction et des Fichiers d'Écriture spéciale uniquement (qui ne sont pas lus), des Fichiers de paramètre et des Matrices (qui sont lus & fermés avant le traitement principal), la condition EOF ne sera jamais établie à vraie et le programme se poursuivra dans une boucle infinie. Pour éviter cette situation, une instruction HALT (conditionnelle) devrait être ajoutée quelque part, de préférence dans la procédure TARGET.
066	VALUE en dehors de la plage des limites du champ	Pour une Champ pour lequel des limites sont définies, les valeurs doivent se trouver dans ces limites. Si la valeur ne se trouve pas dans les limites, ce message s'affiche.
067	Les valeurs initiales ne peuvent pas être utilisées avec	

Nombre	Message	Explication
068	Plage non valide	
069	Maximum de 16 enregistrements dépassé	
070	Trop de valeurs	
071	Trop de champs SORT ou GROUP	
072	Le littéral d'en-tête est plus long que 29 caractères	
073	La valeur est plus longue que la taille du champ	Une valeur est assignée à un Champ dont la taille est trop petite pour la valeur.
074	Structure #1 fermée automatiquement	Si l'instruction END-FOR est rencontrée, les instructions IF ou CASE qui n'ont pas été fermées proprement seront fermées automatiquement.
075	---	
076	Le champ BASED ON doit être défini dans	
077	Nom non unique dans le Dictionnaire	Le nom de l'objet à ajouter existe déjà dans le Dictionnaire technique. Chaque objet dans le Dictionnaire technique doit avoir un nom unique. Dès que la commande "DUPLICATES" est utilisée au début des instructions de maintenance, cet avertissement est affiché parce que le nom sera renommé automatiquement. Ce message est toujours associé au message suivant qui dit que le nom a été modifié.
078	Nom modifié en	Nouveau Nom de l'objet Dès que la commande "DUPLICATES" est utilisée au début des instructions de maintenance, tous les noms non uniques sont renommés par le Générateur MetaSuite. Ce message vous fournira le nouveau nom de l'objet.
079	Usage de mot clé système non valide	
080	Fichier de Dictionnaire #1: #2	
081	Usage du mot COBOL réservé non valide	
082	Les limites ne peuvent pas être utilisées avec	
083	Les valeurs d'égalisation ne peuvent être utilisées avec	
084	L'index doit se référer à un champ	
085	Taille de champ non valide	Certains types de données attendent uniquement une taille de Champ limitée. Ce message est retourné au cas où il y a un conflit de taille de champ.
086	Déjà défini dans	

Nombre	Message	Explication
087	La taille de bloc doit correspondre à	
088	---	
089	Masque de date non valide	
090	Commande de dictionnaire non valide	Ce message est retourné si la première commande d'une instruction de maintenance n'est pas correcte.
091	Type de définition non valide	Le type d'objet sur lequel l'action principale de la commande de maintenance doit être effectuée n'est pas valide. Une commande de maintenance doit toujours être composée de: Action à réaliser Object_type Object_name Object_Characteristics Avec Object_type égal à [FILE RECORD FIELD INDEX LINK]
092	Commande non valide	Ceci est un message général retourné si une commande dans une procédure (Procédure de Fichier Source / Procédure de Fichier Cible / Procédure de Rapport) n'est pas correcte. Des informations plus détaillées concernant ce qui n'est pas correct sont fournies dans les messages retournés précédemment.
093	La taille de l'élément couvrant n'est pas assez grande	La taille du "Nom de l'objet" n'est pas assez grande par rapport aux Champs définis ci-dessous. Le "Nom de l'objet" peut être le nom d'un Fichier, d'un Enregistrement ou d'un Champ contenant des Sous-champs. Veuillez vérifier et corriger les tailles.
094	Type de champ non valide	Le type de données d'un Champ n'est pas correct.
095	La position du Bit excède la taille d'un caractère	
096	Les positions décimales dépassent la taille du champ	S'il y a un problème avec l'option SIZE dans votre définition de Champ, l'indication DECIMAL ne sera plus correcte et ce message sera retourné.
097	Nom de module trop long	
098	Doit être une Procédure Initiale de Fichier entrant	
099	Égalisation plus longue que 57 caractères	
100	Nom d'enregistrement attendu	

Messages 101 à 150

Nombre	Message	Explication
101	Option STANDARD ou OMITTED requis	
102	Les libellés Utilisateur ne sont pas permis	

Nombre	Message	Explication
103	Clé de tri ou de groupe en double	
104	Le Fichier de contrôle ne précède pas le Fichier contrôlé	
105	Le Fichier doit avoir une clé	Pour certains types de Fichier ou certaines commandes MSL une clé doit être définie sur le Fichier (par exemple, Fichiers indexés, Matrices externes, Fichiers contrôlés). Ce message est retourné si aucune clé n'est définie sur le Fichier. Exemple 1: ADD FILE XXX TYPE INDEX FIX 10 BLOCK 15 Exemple 2: GET ARR-employee-master KEY = EMPLOYEE-NUMBER#00053 Exemple 3: SOURCEFILE DEPARTMENT-ARRAY CONTROLLED BY employee-master \Syntax Error 1 Occurred Near: employee-master KEY DEPARTMENT#06502 \FIG105E 1 Fichier doit avoir une clé Ceci est un exemple d'un Fichier séquentiel qui est en même temps un Fichier contrôlé. La case à cocher <i>Automatique</i> est désélectionnée. D'une part, aucune recherche par valeur de clé ne peut être effectuée parce qu'il s'agit d'un Fichier séquentiel qui n'a pas été lu comme une Matrice externe. D'autre part, une structure <i>Contrôlé par</i> nécessite une recherche par clé. Cette situation est conflictuelle, ce qui provoque ce message d'erreur.
106	La taille d'Enregistrement est plus grande que celle du bloc	La taille du bloc d'un Fichier ne doit pas dépasser la taille d'Enregistrement maximale. Ce message est retourné si la taille du bloc est plus petite que la taille maximale de l'Enregistrement.
107	Taille d'Enregistrement maximale requise	L'instruction ADD FILE n'a pas spécifié la taille d'Enregistrement maximale en utilisant les mots clé FIXED, VARIABLE ou UNDEFINED.
108	SPANNED non permis	Cette erreur se produit sur une instruction ADD FILE si l'option SPANNED est spécifiée, mais qu'elle n'est pas autorisée.
109	Mot clé attendu	Il y a un mot clé obligatoire manquant dans l'instruction.
110	Numérique ou littéral attendu	Dans l'instruction, une valeur constante est spécifiée qui ne suit pas les standards suivants: une valeur sans guillemets doit être une valeur numérique, une valeur alphanumérique doit être placée entre guillemets simples.
111	Littéral attendu	
112	Doit être inférieur à 100	
113	Numérique ou nom de Champ attendu	

Nombre	Message	Explication
114	Nom d'Enregistrement ou nom de Champ attendu	L'instruction ADD FIELD doit spécifier un Champ propriétaire connu ou un Enregistrement propriétaire connu. L'objet propriétaire spécifié dans l'instruction ADD FIELD n'est pas un Enregistrement ou un Champ connu.
115	doit être défini dans #1	
116	Le champ SORT est trop large	La taille maximale du champ SORT est de 256 caractères.
117	Ne peut pas dépasser la taille de la page	
118	Caractère d'édition non valide	Ce message est retourné si un masque d'édition est spécifié et qu'il contient un caractère non valide. Cela peut arriver si un format de date donné contient un caractère non valide.
119	Position du caractère d'édition non valide	Ce message est retourné si le masque d'édition du Fichier est incorrect. Veuillez vérifier le masque d'édition du Champ.
120	La position du signe n'est pas valide	
121	Le signe de fin doit être suivi d'un guillemet	Si le masque d'édition d'un Champ numérique ne contient pas de symbole de signe de fin au début du masque d'édition, le signe doit être suivi par le guillemet de fin pour ce masque. Si cela n'est pas le cas, ce message sera retourné.
122	Les caractères les plus à droite sont tronqués	Ce message est retourné si le masque d'édition pour un Champ de travail alphanumérique est plus petit que la taille spécifiée pour le Champ de travail. Il s'agit d'un message d'avertissement vous informant que la taille du masque d'édition (qui est plus petite que celle du Champ de travail) sera respectée.
123	Les chiffres les plus significatifs ont été tronqués	Ce message est retourné si la taille d'un Champ n'est pas assez grande pour les valeurs assignées au Champ (par exemple limite minimale, limite maximale, valeur initiale). Ce message est également retourné si le masque d'édition pour un Champ de travail numérique est plus petit que la taille spécifiée pour le Champ de travail. Il s'agit d'un message d'avertissement vous informant que la taille du masque d'édition (qui est plus petite que celle du Champ de travail) sera respectée.
124	Les chiffres les moins significatifs ont été tronqués	Ce message est retourné si le masque d'édition pour un Champ de travail numérique utilisant des décimales ne contient pas une partie décimale. Il s'agit d'un message d'avertissement vous informant que le masque d'édition (sans aucune précision donnée) sera respecté.
125	Espace non permis	
126	Nom de fichier requis	L'Enregistrement à ajouter ne peut pas être ajouté au Fichier car aucun Fichier n'est actif. Ceci peut arriver si l'instruction 'ADD FILE' précédente a échoué et qu'aucun FILE propriétaire n'a été spécifié pour l'Enregistrement.

Nombre	Message	Explication
127	Nom d'enregistrement requis	S'il n'y a pas d'Enregistrement courant, tous les Champs doivent spécifier un Enregistrement propriétaire. Ce message est retourné si cela n'est pas le cas.
128	Position de champ requise	Aucune POSITION n'est spécifiée pour le Champ. Si l'Enregistrement du Champ est connu, cela ne pose pas de problèmes. Exemple: ADD FIELD C TYPE MIXED
129	La valeur est en conflit avec le type de champ	Une valeur est donnée dans l'instruction, mais elle ne correspond pas au type de données attendu. Cela peut arriver sur la valeur initiale d'un Champ alphanumérique qui est établie à une valeur numérique.
130	La valeur est en conflit avec le signe	
131	Taille requise	Aucune SIZE n'est spécifiée dans la définition du Champ.
132	Commande de fichier requise	
133	La commande TARGETFILE est attendue	
134	La commande BEGIN manque	Ce message est retourné si le Générateur MetaSuite attend une commande pour commencer avec le mot clé BEGIN (pour toutes les définitions de procédure), mais que cette commande n'est pas présente.
135	Commande de procédure manquante	
136	La limite du nom interne est dépassée de #1	
137	Impossible de définir un champ dépendant	
138	Liste incomplète	Ce message est retourné si une commande attend une liste d'objets mais que cette liste d'objets n'est pas complète selon la syntaxe.
139	Entrée de liste attendue	
140	Le maximum de 32 valeurs est atteint	Ce message est retourné si plus de 32 arguments sont fournis pour la commande INVOKE ou DEBUG.
141	Le champ est non défini	Dans le programme MetaSuite, le Champ xxx n'est pas défini ni comme Champ Source, ni comme Champ global, ni comme Champ Cible.
142	Le champ doit être en sous-script	Ce message est retourné si un Champ Source appartient à un Enregistrement Source à occurrence multiple, ou si un Champ global ou un Champ Cible est un Champ à occurrence multiple, et qu'aucun sous-script n'est donné quand le Champ est référencé.
143	Le Sous-script dépasse les occurrences	Ce message est retourné pour les Champs qui doivent être en sous-script.
144	Sous-script(s) additionnel(s) requis pour	Ce message est retourné pour les Champs qui doivent être en sous-script.

Nombre	Message	Explication
145	Trop de sous-scripts pour	Ce message est retourné pour les Champs qui doivent être en sous-script.
146	Un sous-script ne peut pas contenir d'expression.	Ce message est retourné pour les Champs qui doivent être en sous-script.
147	Le sous-script doit être numérique	Ce message est retourné pour les Champs qui doivent être en sous-script.
148	Un sous-script ne peut pas être en sous-script	Ce message est retourné pour les Champs qui doivent être en sous-script.
149	Le sous-script doit être plus grand que zéro	Ce message est retourné pour les Champs qui doivent être en sous-script, mais pour lesquels la valeur du sous-script est zéro ou non définie.
150	Sous-script attendu	Ce message est retourné pour les Champs qui doivent être en sous-script.

Messages 151 à 200

Nombre	Message	Explication
151	Sous-script incomplet	Ce message est retourné si un Champ global ou un Champ Cible est un Champ à occurrence multiple, et qu'aucun sous-script n'est donné quand le Champ est référencé.
152	Première opérande non trouvée	
153	Virgule ignorée	
154	Entrée attendue non trouvée	Ce message est retourné si une commande n'est pas construite correctement (il manque une entrée). Vérifiez votre commande pour détecter l'entrée manquante.
155	Nom de champ système non valide	Ce message est accompagné de <i>SYNTAX ERROR n OCCURED NEAR: xxx</i> où xxx indique quel opérande est attendu comme Champ système, mais qu'il s'agit d'un nom de Champ système non valide.
156	Nom de champ système attendu	Ce message est retourné si un Champ système est attendu dans une commande, mais que la fin de la commande est atteinte.
157	Le sous-script doit être un nom entier	
158	Un champ système ne peut pas être en sous-script	
159	Sous-script ou expression ignorée.	Ce message est retourné si un sous-script dans un programme MetaSuite n'est pas correctement défini.
160	La liste ne peut pas contenir d'expression	
161	La fonction n'a pas de paramètres	

Nombre	Message	Explication
162	Champ de date attendu	
163	Commande non utilisée dans la Procédure de Fichier Cible	
164	Date non valide	
165	La(es) commande(s) suivant EXIT est (sont) ignoré(es)	Aucune commande suivant la commande EXIT ne sera exécutée.
166	La(es) commande(s) suivant EXCLUDE est (sont) ignoré(es)	Toutes les commandes après la commande EXCLUDE seront ignorées.
167	Le nom doit être unique	
168	Option en conflit avec l'option antérieure :	
169	---	
170	Ligne d'édition non définie	
171	Le Champ de travail ne peut trier qu'un seul fichier	
172	Champ non défini dans le Fichier d'entrée	
173	Numéro de ligne attendu	
174	Ne peut pas être inférieur à zéro	
175	Numéro de ligne doit se trouver dans l'ordre	
176	Information incomplète concernant la ligne d'édition	Ce message est retourné si la fin d'un ensemble logique de commandes MetaSuite n'est pas présente, comme par exemple une ligne DETAIL ou une ligne TOTAL sur un Fichier Cible. Cette erreur est généralement liée à une erreur précédente, ce qui fait que le Générateur MetaSuite ignore la commande de fin de l'ensemble logique. Vous devez d'abord corriger l'erreur précédente.
177	Option non permise	
178	La ligne d'édition ne peut pas contenir d'expression	
179	Nombre de lignes à ignorer attendu	
180	Numéro de position attendu	
181	La position n'est peut-être pas spécifiée	
182	Un numéro de TARGETFILE est attendu	
183	Le Fichier Source n'est pas utilisé	Ce message est retourné si un Fichier Source est utilisé qui n'est pas défini comme étant un Fichier Source ou qui n'est pas correctement défini comme Fichier Source.
184	Le Fichier Cible n'est pas utilisé	

Nombre	Message	Explication
185	Position attendue	
186	Doit être précédé d'une valeur d'édition	
187	Le mot clé AND n'est pas présent	Ce message est retourné si une concaténation est attendue (effectuée par le mot clé "AND"). Ce message est souvent retourné pour une instruction sur laquelle une erreur précédente s'est produite. Corrigez d'abord l'erreur précédente.
188	Conflit avec l'option 'Non auto'	
189	Conflit avec 'Non auto' (aucun champ Groupe/Accum.)	
190	Le maximum de 12 colonnes est dépassé	
191	Le nombre de colonnes maximal est dépassé	
192	Le numéro du TARGETFILE doit être supérieur au précédent	
193	Le numéro du TARGETFILE est supérieur à 99	Ce message est retourné si plus de 99 Fichiers Cible ou plus de 99 Rapports sont ajoutés au programme.
194	Fichier déjà utilisé	
195	Fonction #1 inconnue	
196	Opérateur relationnel attendu	Ce message est retourné si une expression conditionnelle est construite, mais qu'aucun opérateur relationnel n'est présent là où il doit être.
197	Le mot clé TO est requis	
198	Parenthèses non équilibrées	
199	Les opérandes doivent appartenir à la même classe	Ce message est retourné si une comparaison est faite et que les opérandes dans la comparaison ne sont pas compatibles. Veuillez vérifier les types de données des opérandes et corrigez-les si nécessaire.
200	Le reste de l'expression est ignorée	Ce message est retourné si une erreur précédente s'est produite sur l'instruction. Corrigez d'abord l'erreur précédente pour pouvoir vérifier l'expression complète.

Messages 201 à 250

Nombre	Message	Explication
201	---	
202	Dépassement de capacité de la ligne d'édition	

Nombre	Message	Explication
203	Le préfixe doit avoir une longueur de 4 caractères	
204	Aucun Champ défini pour #1	Au moins un Champ doit être défini pour ce Fichier ou cet Enregistrement.
205	Sous-script double/triple non permis	
206	Nombre de clés assemblées non valide	Ce message est retourné si la règle de compatibilité concernant les Fichiers Source n'est pas correcte (les Fichiers Source assemblés n'ont pas le même nombre de Champs assemblés définis).
207	---	
208	La clé assemblée de #1 doit être un Champ de travail ou	
209	Aucun enregistrement défini pour le Fichier	
210	Champ clé non défini	
211	Non accessible à partir de cette procédure	
212	Titre par défaut plus long que la largeur de la page	
213	Le Champ compteur doit être un Champ de travail	
214	IF ne peut pas être subordonné à CASE	
215	Doit être précédé d'une valeur d'affectation	
216	Type de données inconsistent	Ce message est retourné si vous voulez assigner des Champs de type de données incompatible. L'assignation suivante donnerait ce message parce que vous ne pouvez pas assigner une valeur alphanumérique à un Champ numérique.
217	---	
218	Trop de décimales	
219	Concaténation non permise	
220	Le champ doit être numérique	
221	La commande d'exemple n'est pas valide	
222	Opérande attendue	
223	Opérande arithmétique attendue	Chaque opérateur doit être suivi d'un espace. Ce message est retourné si cet espace n'est pas présent, ou s'il manque un opérateur dans une expression arithmétique.

Nombre	Message	Explication
224	L'expression ne peut pas être concaténée	
225	Expression incomplète	Ce message est retourné si une expression arithmétique contient une erreur. Cela pourrait être le cas parce que l'opérateur arithmétique n'est pas reconnu s'il n'est pas suivi d'un espace.
226	Doit être précédé d'un opérateur	
227	Doit être précédé d'un opérande	
228	Procédure non définie	
229	Nom déjà utilisé	
230	L'enregistrement #1 n'est pas défini	
231	Valeur d'édition non valide	
232	Valeur d'assignation attendue	Ce message est retourné si la commande contient le symbole d'assignation (=) et qu'il n'est pas suivi d'une valeur.
233	Libellé de transformation:	Si un programme est généré en mode trace, ce message est retourné pour chaque calcul dans le programme.
234	---	
235	Valeur d'édition attendue	
236	Nom de procédure attendu	
237	Doit précéder les Procédures de Fichier Cible	
238	L'opérande BEGIN n'est pas valide	
239	Type de procédure non valide	
240	Procédure en double	Ce message est retourné si deux procédures sont définies avec le même temps d'exécution sur le même Fichier Source ou le même Fichier Cible. Vous ne pouvez utiliser qu'une seule procédure avec un certain temps d'exécution sur le même Fichier Source ou le même Fichier Cible.
241	Une référence antérieure est erronée	
242	Doit être référencé préalablement	Ce message est retourné si une procédure est définie dans le MSL et qu'elle n'est pas utilisée dans une des autres procédures (la reconnaissance est faite sur la base du nom).
243	La procédure est hors séquence	
244	Doit se trouver dans la plage de 1 à 9	

Nombre	Message	Explication
245	Le nombre de SKIP dépasse la taille de la page	
246	Les valeurs d'édition ne peuvent pas se chevaucher	
247	---	
248	---	
249	---	
250	Le fichier contrôle déjà un autre fichier	

Messages 251 à 300

Nombre	Message	Explication
251	---	
252	---	
253	---	
254	Le Champ doit être un Champ de travail	
255	---	
256	---	
257	---	
258	---	
259	---	
260	Une référence en sous-script n'est pas permise	
261	Doit être un Champ de total ou un Champ de travail	
262	La valeur doit être déplacée vers un Champ de travail	
263	L'ensemble "Titre + En-tête + Fin de page" dépasse #1 lignes	
264	La séquence de nom du champ n'est pas valide	
265	Type de ligne d'édition non permis	
266	Impossible de totaliser le champ	
267	Saut non autorisé sur la ligne de titre 0	

Nombre	Message	Explication
268	Fonction non permise	
269	Fonction ignorée	
270	Fonction incomplète	
271	Opérateur arithmétique non valide	
272	Champ de date non permis	
273	Liste d'opérandes non permise	
274	La taille du littéral dépasse 58 caractères	
275	Fourche de chemin non valide dans	
276	Nom de champ non valide	
277	Mot réservé non permis	
278	---	
279	Littéral de longueur nulle	
280	Nom identifiant COBOL non valide	
281	Nom tronqué à	
282	Clause PICTURE COBOL non valide	
283	Champ "basé sur" non défini	
284	Le nom du fichier EXCLUDE n'est pas valide pour #1	
285	L'Enregistrement ne se trouve pas au niveau le plus bas du chemin	
286	Le champ doit avoir un chemin	
287	Valeur non valide - convertie en espaces	
288	Uniquement permis dans une Procédure de Fichier initiale ou une Procédure de Fichier d'entrée.	La commande mentionnée ne peut être utilisée que dans une Procédure de Fichier Source initiale ou dans une Procédure d'entrée du Fichier Source. Ce message est retourné si la procédure dans laquelle est utilisée la commande n'est pas une Procédure de Fichier Source initiale ou dans une Procédure d'entrée du Fichier Source.
289	Non ajouté	
290	Non supprimé	
291	Option non valide pour ce type fichier de base de données	

Nombre	Message	Explication
292	Niveau de confiance inférieur à 80%	
293	'Chemin de fichier' ou 'Nom de fichier' requis	Ce message est retourné si une commande est émise sur un Fichier Source qui doit connaître un nom d'Enregistrement Source et si ce Fichier Source n'est pas défini avec un Chemin. Vous pouvez émettre la commande directement avec le nom de l'Enregistrement Source, ou vous pouvez ajouter le Chemin au Fichier Source pour que l'Enregistrement Source sur lequel la commande doit être émise soit défini comme Enregistrement d'entrée dans ce Chemin.
294	---	
295	Clause clé requise pour ce fichier	
296	Fonction optionnelle (mot de passe attendu)	
297	Option ignorée pour le champ de travail	
298	Remplacé '/' par 'B'	
299	PATH ou CONTROLLED requis	
300	Nom de table attendu	

Messages 301 à 350

Nombre	Message	Explication
301	Enregistrement #1 non défini dans le chemin du fichier	Cette erreur est retournée si l'Enregistrement référencé par cette phrase n'est pas trouvé dans la déclaration de chemin du Fichier courant. Ce Fichier doit être mis en mémoire tampon et l'Enregistrement mentionné doit faire partie du Fichier de mémoire tampon.
302	---	
303	---	
304	N'est pas permis dans SORT ou dans GROUP	
305	---	
306	Ne peut pas dépasser 30 caractères	
307	#1 ne peut pas contenir de guillemets	
308	L'Enregistrement existe, impossible de modifier le mode	
309	Un lien nécessite des Enregistrements FROM et TO	

Nombre	Message	Explication
310	Les liens FROM et TO de l'Enregistrement sont identiques	
311	Nom de lien attendu	
312	La clause FROM doit précéder la clause TO	
313	Le nom de la zone de stockage est requis pour #1	
314	Nom d'index attendu	
315	Le nom du Champ BASED ON est requis	
316	---	
317	---	
318	Aucune donnée référencée pour #1. Ceci pourrait engendrer des erreurs de compilation.	Cette erreur est retournée si aucune donnée d'un certain Fichier n'est référencée. Le fichier n'est pas utilisé.
319	---	
320	Le Fichier doit être un Fichier d'entrée	
321	---	
322	---	
323	---	
324	---	
325	La taille doit être 2, 4 ou 8	
326	---	
327	OCCURS non permis pour #1	
328	Nom de lien ou d'index attendu	
329	VIA non permis au niveau 1	
330	Tri, Pré-passe ou Extraction requis	
331	---	
332	La Clé ne s'applique pas à ce Fichier	
333	La sécurité de fichier empêche l'utilisation de	
334	Ne fournit pas d'index concernant	

Nombre	Message	Explication
335	N'établit pas de lien vers un Enregistrement de chemin antérieur	Ce message est retourné si une instruction de chemin pour un Fichier Source utilise un lien entre 2 Enregistrements qui n'est pas défini entre ces 2 Enregistrements. Vérifiez comment le lien est défini en MDL (entre quels Enregistrements le lien est défini) et vérifiez comment le lien est utilisé en MSL.
336	N'est pas préalablement défini dans le chemin	
337	La taille doit être dans la plage de 1 à 10	
338	Aucun lien trouvé qui fournit le chemin vers #1	
339	Chemin ambigu (nom du lien nécessaire) pour	
340	Enregistrement VIA (Occurrences) non valide pour #1	
341	Trop de sous-chemins (plus de 15)	
342	Pas permis pour la commande TARGETFILE	
343	Seulement 1 valeur est autorisée dans le Fichier Cible.	
344	Nombre maximal de Champs Cible dépassé	
345	La séquence des Enregistrements doit être inversée	
346	Aucune ligne de Détails/Total définie	Ce message est retourné si aucune ligne de Détails ou de Total n'est définie pour le Fichier Cible, ou si les lignes de Détails ou de Total ne sont pas trouvées à cause d'une erreur de syntaxe précédente dans la définition du Fichier Cible. Vérifiez d'abord si une erreur de syntaxe précédente cause ce problème. Si non, ajoutez, dans MetaMap, un Enregistrement Cible de type "Détails" ou "Total".
347	Non valide pour un Champ non-numérique	
348	'OF Nom d'Enregistrement' requis	
349	Le chemin spécifie deux fois l'Enregistrement associé	
350	Les clés ne sont pas permises pour ce type de fichier	Si l'option KEY est utilisée pour un Fichier Source et qu'elle n'est pas correctement définie, ce message est retourné.

Messages 351 à 400

Nombre	Message	Explication
351	Le Fichier ne peut pas être contrôlé	
352	---	
353	Le Champ doit se trouver dans le Fichier START	
354	Référence ambiguë	
355	Préfixe requis	
356	---	
357	Le nom d'index du Chemin d'accès est requis pour START	Si une commande START avec option KEY est utilisée sur un Fichier Source (utilisé pour des Fichiers Source non relationnels), l'Enregistrement d'entrée dans le Chemin doit être accédé par un index (pour permettre l'accès direct de la commande START). Ce message est retourné si l'Enregistrement d'entrée n'est pas accédé par un index.
358	---	
359	Fin de commande ou OFF attendu	
360	Littéral ou OFF attendu	
361	Le préfixe ne peut pas dépasser les 12 caractères	
362	Les traits d'union consécutifs ne sont pas autorisés	
363	---	
364	---	
365	Aucun DBNAME défini pour	
366	Dépassement de la liste de format dans le Fichier	L'usage du code de la liste de format est réservé pour usage interne. Contactez votre administrateur MetaSuite ou IKAN Solutions N.V.
367	Aucune clé de stockage définie pour	
368	---	
369	---	
370	---	
371	#1 trop long Tronqué à '#2'	
372	---	
373	---	

Nombre	Message	Explication
374	---	
375	---	
376	---	
377	Champ d'occurrence non trouvé dans l'Enregistrement	
378	Le champ DEPENDING ON n'est pas valide	
379	---	
380	---	
381	---	
382	---	
383	---	
384	L'option "rapports automatiques" n'est pas permise dans le Fichier Cible ????se trouve où ????	Si la commande de transfert d'une Procédure d'entrée de l'Enregistrement est utilisée, l'option "rapports automatiques" n'est pas permise dans le Fichier Cible.
385	Non valide dans une Procédure de Rapport initiale triée	
386	---	
387	Conflit de EBCDIC/Multi-enregistrements pour	Les Fichiers à multi-enregistrements contenant des Champs EBCDIC dans plus d'une définition d'Enregistrement ne sont pas supportés par le compilateur MicroFocus COBOL.
388	L'enregistrement doit avoir une Clé de stockage	Ce message est retourné si aucune Clé de stockage n'est retrouvée pour une action spécifique sur un Enregistrement pour lequel une Clé de stockage est requise.
389	Conflit de chevauchement de champ EBCDIC pour	Le chevauchement de Champs EBCDIC dans un Fichier Source EBCDIC n'est pas supporté par un compilateur MicroFocus COBOL. Si le chevauchement de Champs est utilisé dans ce cas, ce message est retourné.
390	---	
391	Définitions en double trouvées pour #1	
392	Le Fichier a été modifié depuis la dernière vérification	
393	Le type de Clé de stockage n'est pas valide pour	
394	Enregistrement non utilisé	
395	Impossible d'exclure un Enregistrement subséquent	
396	Le Champ DEPENDING ON ne peut pas chevaucher #1	

Nombre	Message	Explication
397	Non valide dans une Procédure d'entrée de l'Enregistrement	
398	L'Enregistrement d'entrée doit avoir une Clé de stockage	Ce message est retourné si vous avez un Fichier Source CONTRÔLÉ PAR IDMS pour lequel l'Enregistrement d'entrée n'a pas de Clé de stockage définie. Vous devez ajouter une Clé de stockage à l'Enregistrement d'entrée ou vous devez choisir une autre entrée d'Enregistrement avec une Clé de stockage.
399	Enregistrement DB: Clé de stockage ou nom de base de données non valide	
400		

Messages 401 à 450

Nombre	Message	Explication
401	L'usage de membres multiples non valide dans le lien	
402	---	
403	Nom de champ identique au nom de fichier ou d'enreg.	
404	Chemin d'accès non valide pour le fichier de base de données	
405	Vérification du nom de fichier Datacom en cours	
406	Le Mode d'exécution IMS/ Redémarrable n'est pas supporté par IBM/ODBC	
407	Le fichier Datacom nécessite une Clé d'index de stockage	
408	Aucune ligne TOTAL n'est définie pour imprimer/écrire	
409	Commande non valide dans une procédure Groupe	
410	Format de la Clé de fichier aléatoire non valide	
411	---	
412	---	
413	CONVERT doit avoir exactement 1 détail	
414	---	

Nombre	Message	Explication
415	Pas permis pour la commande CONVERT	
416	TIME doit être un champ numérique	
417	Le type TIMESTAMP doit être CHARACTER ou NUMERIC	Ce message est retourné si un Champ est défini en utilisant l'option "timestamp", mais que le type de données n'est pas correct.
418	Le format de date doit être 'YYYYMMDD'	Il peut arriver que seul le format de date ISO 'YYYYMMDD' est supporté, par exemple pour les Fichiers Source SQL. Dans ces cas, une option de pré-compilation peut être établie pour lire toute date dans le format de date ISO.
419	Données zonées modifiées en décimales SQL	
420	Avertissement concernant le Fichier Source	Un avertissement est généré pour un Fichier Source. MetaSuite utilise cette erreur pour fournir le nom du Fichier sur lequel l'avertissement s'est produit. Le type d'avertissement est affiché sur la ligne suivante.
421	EXEC-SQL n'est pas supporté pour ODBC sur IBM	
422	---	
423	Erreur sur le Fichier Cible	Une erreur s'est produite sur un Fichier Cible. Cette erreur est utilisée par MetaSuite pour fournir le nom du Fichier sur lequel l'erreur s'est produite. Le type d'erreur est affiché sur la ligne suivante.
424	Hogan n'est plus supporté	Depuis la version 7.2, Hogan n'est plus supporté.
425	Les compilateurs Fujitsu ne supportent pas:	Certaines options ne sont supportées que par un nombre limité de compilateurs. Par exemple, le mode EBCDIC sur des Fichiers d'index séquentiels n'est pas supporté par les compilateurs Fujitsu.
426	SYS-RAW non disponible après SORT	Les Champs numériques sont stockés dans un autre format s'ils sont copiés vers un Fichier de tri. Cela est fait pour des raisons de performances. Après l'instruction SORT, la fonction SYS-RAW ne peut plus être utilisée sur ces Champs. Le résultat peut induire en erreur.
427	SQL utilisé dans la Procédure de Programme Initial	Ce message ne s'applique qu'aux systèmes Unix et Windows. Certains dialectes SQL et certaines bases de données ODBC ne supportent pas certaines instructions SQL utilisées immédiatement après le chargement du programme. Si l'option <i>MOVE-PROGINIT-IF-SQL</i> est établie à <i>ON</i> , la Procédure initiale de programme sera exécutée après l'ouverture des Fichiers, et après la Procédure de Fichier initiale.
428	Le champ #1 doit être défini comme champ nullable	Si vous utilisez les mots clé SYS-STATUS et STATUS-INCLUDED dans du SQL imbriqué, le Champ doit être défini comme un Champ "Nullable".

Nombre	Message	Explication
429	Niveaux à occurrence multiple	Les niveaux à occurrence multiple ne sont pas supportés pour les Champs Cible et les Champs de travail. À <i>occurrence multiple</i> signifie: un Champ à occurrence multiple dans un Champ de groupe également à occurrence multiple. (Cela engendre des tables multidimensionnelles.)
430	Comparaison entre dates et non-dates	Tous les Champs de type <i>date</i> seront convertis en Champs numériques de type YYYYMMDD. L'avantage de cette méthode est qu'il est possible de comparer des dates de type caractère avec des dates numériques. Le désavantage est que vous ne pouvez pas comparer des dates de type caractère avec des Champs de type caractère. Vous pouvez contourner cette erreur en utilisant une redéfinition.
431	L'usage de SYS-EDIT dans un champ de date est superflu	Tous les Champs de type <i>date</i> seront convertis en Champs numériques de type YYYYMMDD. Pour cette raison, l'usage de SYS-EDIT pour les Champs de date n'est pas possible.
432	Numéro de construction invalide - le numéro doit être composé de 2 chiffres	CHANGE DEFAULT BUILD: Le Numéro de construction du Dictionnaire est un numéro entre 00 et 99.
433	Le type de données SQL de #1 n'est pas supporté	
434	La taille du type de données SQL de #1 n'est pas valide	
435	---	
436	Numéro de FORMAT attendu	
437	Numéro de FORMAT non valide	
438	Uniquement autorisé pour un champ numérique cumulable	
439	Uniquement autorisé dans une ligne détail éditale	
440	---	
441	Conversion de champs numériques > 8 chiffres impossible	
442	FULL requiert un champ VARCHAR	
443	---	
444	Siècle non valide - doit être composé de 2 chiffres	
445	Aucune table d'Utilisateur définie	
446	Renommer peut affecter les programmes existants	
447	---	
448	---	

Nombre	Message	Explication
449	---	
450	Année de bascule non valide	

Messages 451 à 500

Nombre	Message	Explication
451	POSITION requise pour le champ Flux-entrant	
452	Table non trouvée dans le catalogue SQL	
453	---	
454	---	
455	---	
456	---	
457	---	
458	---	
459	---	
460	---	
461	Une erreur s'est produite dans le module #1	
462	---	
463	Commande de fichier Flux-entrant hors séquence	
464	Le nom du créateur dépasse la limite de 8 caractères	
465	Un créateur est requis pour un enregistrement SQL	
466	'Of' non permis dans une colonne SQL	
467	Le nom de table #1 est mis devant #2	
468	Nom de table attendu	
469	Clause WHERE requise - non trouvée	Pour un Fichier Source SQL contrôlé, une clause WHERE doit être définie dans son Chemin. Ce message est retourné si la clause WHERE n'est pas présente.
470	Le maximum de 12 tables est atteint	
471	---	

Nombre	Message	Explication
472	---	
473	Nom de colonne attendu	
474	---	
475	---	
476	---	
477	---	
478	---	
479	---	
480	---	
481	---	
482	---	
483	La clé n'est pas valide pour un fichier SQL	
484	---	
485	SORT pas permis pour EXEC=IMS	Si un programme est généré avec "IMS" comme exécution par défaut, le tri ne peut pas être effectué ni sur des Fichiers Source non SQL ni sur les Fichiers Cible. Si le SORT est spécifié de toute façon, ce message est retourné.
486	---	
487	OCCURS imbriqué interdit pour les champs Flux-entrant	
488	---	
489	Le champ SYS-SQL-NULL n'est pas nullable	
490	Le champ SYS-SQL-LENGTH n'est pas variable	
491	---	
492	---	
493	---	
494	---	
495	---	
496	---	
497	---	
498	---	
499	Nom de schéma attendu	Dans une définition de Fichier IDMS, un nom de schéma doit être ajouté. Ce message est retourné si le nom du Schéma n'est pas le nom attendu par le Générateur MetaSuite.

Messages 501 à 550

Nombre	Message	Explication
501	---	
502	Option non permise pour ce fichier	
503	---	
504	---	
505	---	
506	Impossible de charger la table	
507	---	
508	Taille d'Enregistrement ou de Bloc invalide pour COBOL	Ce message est retourné si la taille de l'Enregistrement ou la taille du bloc est trop grande pour un Fichier, ou si la taille de l'Enregistrement est trop grande pour un Enregistrement.
509	---	
510	Aucune commande GET pour le fichier contrôlé:	
511	---	
512	Mot clé SQL attendu	
513	Mot clé, variable Host ou nom attendu	
514	Clause INTO attendue	
515	---	
516	Variable Host attendue	
517	Instruction SQL trop longue	
518	Liste des champs ou clause INTO attendue	
519	Erreur de syntaxe SQL détectée par DBMS	
520	Avertissement de syntaxe SQL détecté par DBMS	
521	Nombre maximal de host variables "définition" dépassé	
522	Nombre maximal de host variables "programme" dépassé	
523	---	
524	---	
525	---	

Nombre	Message	Explication
526	L'instruction SQL DDL n'est pas supportée	
527	Le mot clé DIALECT est attendu	
528	Dialecte SQL non valide spécifié	
529	---	
530	---	
531	---	
532	---	
533	---	
534	Un seul dialecte peut être spécifié	
535	Nom de curseur attendu	
536	---	
537	Nom de curseur attendu	
538	Nom de curseur non valide	
539	Curseur non déclaré	
540	Nombre maximal de curseurs dépassé	
541	Le mot clé OF est attendu	
542	Le mot clé END-EXEC est attendu	
543	Non valide dans des procédures communes	
544	---	
545	---	
546	La clause VALUE est manquante pour le champ 88 #1	
547	Le mot clé NOT peut uniquement précéder un niveau 88	
548	---	
549	---	
550	Un seul titre permis pour Oracle	

Messages 551 à 600

Nombre	Message	Explication
551	---	
552	---	
553	Table des limites système non valide	
554	Appelez le support technique	
555	---	
556	---	
557	Le résumé doit contenir une ligne TOTAL	
558	PREPARE doit être ON ou OFF	
559	---	
560	Un seul En-tête requis (Sybase)	
561	L'option 'NOAUTO heading' n'est pas permise (message obsolète)	
562	Dialecte SQL non spécifié	Ce message est retourné si vous voulez générer du MGL pour un programme utilisant un Fichier Source SQL, et qu'aucun dialecte SQL n'est spécifié dans le Générateur MetaSuite.
563	Instruction IMPLEMENT: dépassement de capacité de la table	
564	Le masque d'édition Utilisateur n'est pas utilisé pour:	Ce message s'affiche si un masque d'édition superflu est spécifié pour un Champ. Ce masque d'édition ne sera pas utilisé et le Générateur renvoie un avertissement à l'Utilisateur.
565	Variable non définie	Il s'agit d'un message MTL. La variable MTL utilisée dans cette expression n'est pas encore définie par une instruction SET.
566	Nombre maximal de variables atteint	Il s'agit d'un message MTL. Le nombre de variables MTL dépasse la limite. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
567	Une variable protégée ne peut pas être modifiée	Il s'agit d'un message MTL. Une variable protégée ne peut pas être modifiée par l'instruction SET.
568	L'instruction IMPLEMENT n'est pas permise dans une table MRL	
569	Erreur de syntaxe dans l'expression FOR	Il s'agit d'un message MTL. Pour plus d'informations concernant l'instruction FOR, se référer au chapitre Langage de modèle MetaSuite (page 99).

Nombre	Message	Explication
570	Message MTL général	Il s'agit d'un message MTL, généré par la commande MSG, WARNING ou ERROR. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
571	Libellé manquant	Il s'agit d'un message MTL. Le libellé spécifie dans l'instruction "GOTO" ou "SKIP" n'est pas trouvé. Pour plus d'informations concernant les instructions de saut, se référer au chapitre Langage de modèle MetaSuite (page 99).
572	THEN/ELSE/END-IF sans IF	Il s'agit d'un message MTL. L'instruction THEN, ELSE ou END-IF est rencontrée sans l'instruction correspondante IF appartenant au même niveau. Raisons possibles: une instruction de saut dans une structure IF, une combinaison de différentes structures non valides, une codification de l'instruction IF oubliée.
573	Le libellé est plus long que 8 caractères	Il s'agit d'un message MTL. Le nom d'un libellé ne peut pas être plus long que huit caractères.
574	END-EVALUATE ou END-IF manquant	Il s'agit d'un message MTL. Une structure IF doit être terminée par une instruction END-IF. Une structure EVALUATE doit être fermée par l'instruction END-EVALUATE. Ce message est retourné si cela n'est pas le cas. Pour plus d'informations concernant les structures IF et EVALUATE, se référer au chapitre Langage de modèle MetaSuite (page 99).
575	EVALUATE correspondant manque	Il s'agit d'un message MTL. L'instruction WHEN, THRU ou END-EVALUATE est rencontrée sans l'instruction correspondante EVALUATE appartenant au même niveau. Raisons possibles: une instruction de saut dans une structure EVALUATE, une combinaison de différentes structures non valides, une codification de l'instruction EVALUATE oubliée.
576	EVALUATE/IF - Pile complète	Il s'agit d'un message MTL. Le nombre d'imbrications dépasse la limite de 16. Pour plus d'informations concernant les structures IF et EVALUATE, se référer au chapitre Langage de modèle MetaSuite (page 99).
577	Erreur de syntaxe dans l'instruction MTL	Erreurs de syntaxe MTL générales Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
578	Champ numérique comparé avec un champ non-numérique	Dans les structures IF, les expressions numériques et les expressions non-numériques ne peuvent pas être comparées entre elles. Pour plus d'informations concernant les instructions IF, AND et OR, se référer au chapitre Langage de modèle MetaSuite (page 99).
579	---	
580	Division par zéro	Il s'agit d'un message MTL. Instruction SET: la division par zéro n'est pas autorisée. Pour plus d'informations concernant l'instruction SET, se référer au chapitre Langage de modèle MetaSuite (page 99).

Nombre	Message	Explication
581	Dépassement de capacité de ligne de commandes	Il s'agit d'un message MTL. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
582	FOR/NEXT - Pile complète	Il s'agit d'un message MTL. Le nombre d'imbrications dépasse la limite de 16. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
583	La valeur de STEP n'est pas numérique	Il s'agit d'un message MTL. Le résultat d'une expression STEP dans une boucle FOR doit être numérique. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
584	La valeur TO n'est pas numérique	Il s'agit d'un message MTL. Le résultat d'une expression TO dans une boucle FOR doit être numérique. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
585	La valeur FOR n'est pas numérique	Il s'agit d'un message MTL. Le résultat de l'expression FOR dans une boucle FOR doit être numérique. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
586	---	
587	NEXT sans FOR	Il s'agit d'un message MTL. L'instruction NEXT est rencontrée sans l'instruction correspondante FOR appartenant au même niveau. Raisons possibles: une instruction de saut dans une structure FOR, une combinaison de structures différentes non valide, une codification de l'instruction FOR oubliée, ou une instruction FOR ajoutée dans une instruction REMARK. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
588	FOR sans NEXT	Il s'agit d'un message MTL. L'instruction "FOR" est rencontrée sans l'instruction correspondante "NEXT" appartenant au même niveau. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
589	La valeur STEP est zéro	Il s'agit d'un message MTL. Le résultat d'une expression STEP dans une boucle FOR ne peut pas être zéro. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
590	Les caractères ne sont pas permis.	Il s'agit d'un message MTL. Les caractères ne peuvent pas faire partie d'une expression numérique. Si vous voulez utiliser des variables dans une expression numérique, veuillez les placer entre crochets droits. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
591	OR/AND sans IF	Il s'agit d'un message MTL. L'instruction OR ou AND est rencontrée sans l'instruction correspondante IF appartenant au même niveau sur une ligne précédente. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).

Nombre	Message	Explication
592	Commande THRU sans WHEN	Il s'agit d'un message MTL. L'instruction THRU est rencontrée sans l'instruction correspondante WHEN sur la ligne précédente. Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
593	Le mot réservé OTHER n'est pas utilisé correctement	Il s'agit d'un message MTL. Le mot réservé <i>OTHER</i> est utilisé en dehors d'un bloc d'instruction <i>EVALUATE</i> , ou le mot réservé se trouve à un endroit non valide. (Par exemple dans une expression THRU) Pour plus d'informations, se référer au chapitre Langage de modèle MetaSuite (page 99).
594	Fonction MTL non valide	
595	Erreur de syntaxe dans la commande de fonction	
596	Fonction de chaîne de caractères - position non valide	
597	Niveau de CALL supérieur à 16	
598	La table de Dictionnaire référencée est manquante ou non valide	
599	Option FREEZE/UNFREEZE non valide	
600	Avertissement! Clé de licence périmée	Veuillez contacter votre Administrateur MetaSuite ou IKAN Solutions N.V. pour obtenir et installer la nouvelle Clé de licence.

Messages 601 à 650

Nombre	Message	Explication
601	La valeur pour Null sera ignorée	Premier cas: Certains types de Champs tels que BINARY et FLOATING-POINT ne supportent pas l'usage d'indicateurs Null. Le paramètre pour Null sera ignoré. Deuxième cas: Les valeurs pour Null ne peuvent pas être utilisées sur les Fichiers Cible de type "Séquentiel contrôlé" à cause de redéfinitions potentiellement confuses.
602	Les champs Null à occurrence ne sont pas autorisés.	Actuellement, cette option n'est pas supportée. Un Champ avec DBNAME INNULL, OUTNULL ou OUTNULR ne peut pas être un Champ à occurrence multiple.

Nombre	Message	Explication
603	Le champ NonNull n'a pas d'indicateur Null	Cet avertissement est affiché si vous assignez un SYS-NULL-VALUE au SYS-STATUS d'un Champ qui ne contient pas d'indicateur Null. En d'autres termes: le paramètre DBNAME et le paramètre SYS-NULL sont en conflit. Le SYS-STATUS d'un Champ peut être établi comme SYS-NULL-VALUE, même si le Champ contient un indicateur non Null (DBNAME est NOTNULL). Le Champ sera initialisé par cette action!
604	Littéral hexadécimal non valide	
605	Le type du champ de tri n'est pas valide	Cette erreur se produit si un Champ de tri est de type FLOAT ou BIT. Ces types de Champ peuvent générer des résultats de tri incorrects à cause des limitations du processus de tri.
606	Le dialecte SQL est établi à ODBC	Ce message est retourné si l'instruction SET SQL DIALECT est utilisée pour un dialecte non ODBC tandis que l'environnement COBOL utilisé est Fujitsu. La commande SET SQL DIALECT est annulée.
607	Type de champ non valide spécifié	
608	La clé de matrice n'est pas égale à la clé de tri	Il est évident qu'une matrice externe contenant une clé de recherche ne peut pas être triée en utilisant une autre clé. Si non, l'instruction SEARCH donnera des résultats imprévisibles.
609	La Clé de tri dans l'arborescence doit être croissante	Pour une matrice externe avec clé de recherche dans MetaSuite, la clé de recherche est générée en ordre ascendant. Si vous effectuez un tri initial vers la matrice, l'ordre de la première clé (qui doit être la même clé que la clé de recherche) doit être ascendant.
610	Les champs à occurrence multiple ne sont pas autorisés.	Si l'option CODE-CONTROL est spécifiée pour un type d'Enregistrement délimité, aucun champ à occurrence multiple ne peut être inclus dans la description de l'Enregistrement.
611	Les champs redéfinis ne sont pas autorisés.	Si l'option CODE-CONTROL est spécifiée pour un type d'Enregistrement délimité, aucun champ redéfini ne peut être inclus dans la description de l'Enregistrement.
612	Aucun SORT ou ORDER BY trouvé pour la clé de matrice	Aucun SORT ou ORDER BY n'est trouvé lors de la lecture du Fichier de matrice externe dans l'arborescence binaire. Si la séquence de lecture n'est pas correcte, la matrice externe ne fonctionnera pas comme prévue. Aucun SORT ou SQL ORDER BY n'est trouvé non plus. Si l'Utilisateur est sûr que l'ordre de lecture sera correct, cet avertissement est superflu.
613	Table de Code-Contrôle standard refusée	Ce message s'affiche si le paramètre CODE-CONTROL est spécifié pour un type de Fichier qui utilise déjà un type de Table de Code-Contrôle spécifique. Il s'agit d'un simple avertissement. Les types spécifiques sont pour le moment DELIMITED et XML.

Nombre	Message	Explication
614	Nom d'enregistrement attendu	La phrase Fichier Cible/Détails contient un paramètre RECORD dans lequel le nom de l'Enregistrement peut être spécifié. Ce message est retourné si le nom est manquant.
615	Colonne non trouvée dans l'instruction de chemin	La colonne (son préfixe) référée n'est pas spécifiée dans l'instruction de chemin précédente.
616	---(le mode EXEC est IMS)	
617	Procédure de Saisie d'Enregistrements multiples non supportée	Cette erreur est retournée si vous définissez une procédure d'entrée pour un Enregistrement qui peut avoir une occurrence multiple dans un chemin.
618	Table de Code-Contrôle non définie	La Table de Code-Contrôle, spécifiée dans la commande ADD FILE, ne peut pas être trouvée dans le Dictionnaire. La Table de Code-Contrôle par défaut sera utilisée. Pour plus d'informations concernant les Tables de Code-Contrôle, se référer à la section Tables de Code-Contrôle (page 65).
619	Le dialecte SQL devrait être ODBC	ODBC est le seul dialecte SQL supporté par MetaSuite pour ce compilateur.
620	Paramètre de guillemets non valide	La syntaxe correcte est la suivante: "CHANGE DEFAULT QUOTE TO valeur", où la valeur peut être SINGLE, DOUBLE ou ORIGINAL.
621	L'usage de SYS-STATUS n'est pas autorisé pour #1	Les Champs de fonction ne contiennent pas de Champ de statut. Pour cette raison, l'usage de SYS-STATUS n'est pas autorisé pour les Champs de fonction.
622	Champ de fonction: réinitialiser à NonNull	Les Champs de fonction ne contiennent pas de Champ de statut. Pour cette raison, la définition de "INNULL", "OUTNULL", ou "OUTNULR" est réinitialisée à "NOTNULL".
623	---	
624	Paramètre de point décimal non valide	CHANGE DECIMAL POINT TO COMMA ou POINT. Toute autre valeur produira cette erreur.
625	La taille du Champ n'est pas correcte	La taille du Champ ne correspond pas au masque d'édition ou au masque de date.
626	Les Enregistrements à occurrence multiple ne sont pas possibles.	Pour certains types de Fichiers tels que les Fichiers délimités, il est impossible de définir des structures à Enregistrements multiples, parce que vous devez savoir quel type d'Enregistrement vous allez analyser avant de l'analyser. Cependant, dans le cas des structures à Enregistrements multiples, vous ne connaissez le type d'Enregistrement qu'après avoir lu l'Enregistrement.
627	Index non valide (zéro ou trop élevé)	Référence non validé vers un Fichier, un Enregistrement ou une variable de Champ MTL.
628	Ce préfixe est réservé	Les préfixes <i>FILE-</i> , <i>RECORD-</i> et <i>FIELD-</i> sont réservés. Définissez les variables MTL en utilisant d'autres préfixesL
629	Nom de variable spécial non valide	Les préfixes "FILE-", "RECORD-" et "FIELD-" sont réservés. La variable MTL à laquelle vous référez n'est pas prise en charge.

Nombre	Message	Explication
630	Action SQL non valide	L'action SQL peut être INSERT, UPDATE ou DELETE.
631	Erreur de syntaxe IDMS: Attendu	Expression non valide. Le vérificateur de syntaxe IDMS attend le mot clé spécifié (ou un des mots clé spécifiés).
632	Double point non autorisé dans un nom de table	Un double point (:) ne peut être utilisé dans un nom de table. MetaSuite ne prend pas encore en charge cette fonctionnalité.
633	Un Masque d'édition est requis pour les champs PRN	Pour chaque Champ PRN, un masque d'édition doit être défini.
634	Aucune correspondance trouvée pour la Clé de Contrôle	La Clé de Contrôle n'est pas trouvée.
635	Avertissement! Fourche de chemin utilisé dans	Le Champ clé du Fichier contrôlant est un Champ de travail. Tester sur des "chemins fourchés" est inutile dans ce cas. L'utilisateur est responsable de la définition correcte du Champ dans la Procédure de Fichier initiale de contrôle.
636	Aucun espace réservé pour l'octet OUTNULR	Ce Champ est défini comme "OUTNULR". Cela signifie qu'à droite du Champ un octet doit être réservé pour le caractère Null. Cet espace n'est pas disponible.
637	Aucun espace réservé pour l'octet OUTNULL	Ce Champ est défini comme "OUTNULL". Cela signifie qu'à gauche du Champ un octet doit être réservé pour le caractère Null. Cet espace n'est pas disponible.
638	Le Sous-chemin dépend du chemin occurrent	Ce type de structure n'est pas encore supporté.
639	Champ Cible ou Champ de travail non valide	Un Champ Cible ou un Champ de travail ne peut pas être du type 'BIT'. Non encore supporté.
640	L'expression ne peut pas être interprétée:	L'expression MTL ne peut pas être validée parce que des types de variables MTL différents ne peuvent pas être comparés entre eux. Les données numériques et les données non-numériques ne peuvent pas être comparées les unes avec les autres.
641	Parenthèses pas encore fermés	Par exemple: Les parenthèses de la clause ORDER doivent être fermées avant de démarrer la clause WHERE.
642	Parenthèses fermées trop tôt	Les parenthèses pour la clause SELECT/WHERE sont fermées trop tôt.
643	Le champ #1 devrait être alphanumérique	Les fonctions telles que SYS-TRIM et SYS-UPPERCASE ne peuvent être utilisées que pour les Champs alphanumériques et ne peuvent pas être utilisées pour les Champs numériques.
644	Une accumulation double peut se produire	Si vous combinez des lignes de Détails et des lignes de Total dans un même rapport, la logique d'accumulation pourrait être effectuée deux fois. Soyez prudent dans ce cas.
645	DEPENDING ON non permis. Motif:	Les tables ayant un nombre variable d'éléments ne sont pas permises dans ce cas. Par exemple, dans le cas de Fichiers de fonction, il n'est pas permis d'avoir des Champs à occurrence multiple avec une occurrence variante.

Nombre	Message	Explication
646	La structure imbriquée #1 n'est pas permise	Cette option n'est pas encore supportée.
647	Le Code fin de rangée fait partie du Délimiteur	L'Utilisateur a spécifié un Fichier délimité contenant des Séparateurs de colonnes (ou des délimiteurs) et un Code fin de rangée. Pour éviter la confusion, le Code fin de rangée ne peut pas faire partie du Séparateur de colonnes.
648	Aucune variante fixe de la table de codes	La table de contrôle de sortie #1 est utilisée pour les Fichiers Cible variables. La table de contrôle de sortie correspondante pour les Fichiers Cible n'est pas définie.
649	La clé de matrice #1 n'appartient pas à #2	La clé de matrice #1 doit faire partie de la matrice #2.
650	Le préfixe n'est pas supporté pour les fichiers XPC	XPC = XML avec Contrôle de Chemin. Dans la version actuelle, les Fichiers XML avec contrôle de chemin sont supportés pour les Fichiers Source sans préfixe de fichier.

Messages 651 à 700

Nombre	Message	Explication
651	La valeur de la clé n'est pas égale à la longueur de la clé	La clé de la matrice externe et la valeur de la clé à laquelle elle est comparée n'ont pas la même taille.
652	Le nombre maximal de valeurs clé est #1	
653	Masque de date numérique requis pour #1	
654	Commande GET: plus de clés que de valeurs	
655	Le champ d'index #1 est trop petit	Un des Champs d'index d'un Champ à occurrence multiple est trop petit pour le nombre de Champs qu'il peut contenir.
656	La fonction #1 n'est pas supportée dans IF/CASE	Cette fonctionnalité n'est pas toujours supportée dans un bloc IF ou dans un bloc CASE.
657	#1 faiblement pris en charge dans IF/CASE	Cette fonctionnalité n'est pas toujours supportée dans un bloc IF ou dans un bloc CASE. Veuillez contacter le service de support MetaSuite.
658	Le nombre maximal de valeurs Total est #1	

Nombre	Message	Explication
659	Valeur d'argument non valide	Il y a plusieurs possibilités: 1. (Module XML formaté) Le mot clé XML-NAME est utilisé, mais aucun nom XML n'est fourni. 2. (Module XML formaté) Le mot clé XML-TYPE est utilisé, mais aucun type XML n'est fourni ou le type XML fourni n'est pas valide (les valeurs valides sont "ATTRIBUTE" ou "GROUP"). 3. Le deuxième paramètre de la fonction SUBSTRING est non valide ou omis. 4. Le troisième paramètre de la fonction SUBSTRING est non valide ou il a été omis.
660	Masque de formatage non valide ou manquant	Il y a plusieurs possibilités: 1. (Module XML formaté) La valeur d'un mot clé (XML-PATH, FORMAT-MASK, XML-DECLARATION) avec une valeur de propriété longue (250 caractères) n'est pas spécifiée. 2. (Module Unicode) Une entrée dans la table de Dictionnaire UCD001 n'est pas valide (CCSID à cinq positions suivi de '-', 'E', 'A' ou 'U').
661	---	
662	Instruction COMPUTE trop grande (> 20 lignes)	
663	Le nombre de Fichier maximal est atteint.	Le Modèle MetaMap contient plus de 99 Fichiers Source.
664	Le nombre maximal d'enregistrements est atteint	Le Modèle MetaMap contient plus de 99 Enregistrements Source.
665	CCSID non valide (Fichier, enregistrement, champ)	Il y a plusieurs possibilités: 1. (Module Unicode) Le modèle MetaMap contient un CCSID non valide (inférieur à zéro ou supérieur à 65535 - X'FFFF') au niveau du Fichier, de l'Enregistrement ou du Champ. 2. (Module Unicode) Le Modèle MetaMap contient le mot clé CCSID, mais aucun CCSID n'est fourni. 3. (Module Unicode) Le Modèle MetaMap contient le mot clé CCSID, mais le CCSID fourni avec ce mot clé ne se trouve pas dans la table de Dictionnaire UCD001.
666	#1: #2 CCSID #3 non valide pour le type #4	Il y a plusieurs possibilités: 1. (Module Unicode) Un CCSID non valide (un CCSID non Unicode) est fourni pour un Champ de données Unicode. 2. (Module Unicode) Plus de 3650 CCSIDs doivent être sauvegardés dans la table ESYM-2B-ADDED. 3. (Module Unicode) Tous les CCSIDs utilisés dans un modèle MetaMap doivent être déclarés dans la table de Dictionnaire UCD001.

Nombre	Message	Explication
667	#1: CCSID #2 non valide pour le type #3	La table de Dictionnaire UCD001 déclare les CCSIDs valides et leur type (ASCII, EBCDIC, UNICODE). Tous les CCSIDs mentionnés dans le Modèle MetaMap doivent être utilisés selon ces règles. Par exemple: CORRECT = un CCSID Unicode pour un Champ de données Unicode INCORRECT = un CCSID ASCII pour un Champ de données Unicode (Module Unicode)
668	La fonction #1 n'est pas permise	Une fonction doit avoir un nom (un nom MetaSuite ou un nom défini par l'Utilisateur). Les fonctions MetaSuite valides sont: ABSOLUTE-VAL, ASCII, ASCII-UNICODE, ASCII-UTF8, BINARY, CURRENT-DATE, EBCDIC, EBCDIC-UNICODE, EBCDIC-UTF8, HEXADECIMAL, INTEGER-OF-DATE, LOWER-CASE, NUMVAL, REVERSE, SYS-LENGTH-R, TRIM, UNICODE-ASCII, UNICODE-EBCDIC, UPPER-CASE, UTF8-ASCII, UTF8-, EBCDIC, WHEN-COMPILED
669	Séparateur de colonnes (1-3 caractères) non valide	Le mot clé COLUMN-SEPARATOR d'un Fichier délimité est non valide (exemple valide: COLUMN-SEPARATOR ';').
670	Code fin de rangée (1-3 caractères) non valide	Le mot clé ROW-TERMINATOR d'un Fichier délimité est non valide (exemple valide: ROW-TERMINATOR '\')
671	Date non valide. Le format de date doit être YYYYMMDD	Lors de l'attribution d'une valeur de date à SYS-DATE, l'Utilisateur doit veiller à ce que le format de cette valeur de date soit spécifié ainsi YYYYMMDD.
672	#1-Parenthèse #2 manquante	La syntaxe d'une fonction mentionne la séquence des paramètres obligatoires et optionnels. Les paramètres doivent être entourés par des parenthèses à gauche et à droite. Ce message peut s'afficher si le nombre de paramètres et leurs virgules de séparation ne sont pas valides. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" #2 est la position des parenthèses, par exemple "Ouverture" Comme dans "INSTRING-Parenthèse Ouverture manquante"
673	#1- Paramètre incorrect #2	La syntaxe d'une fonction mentionne la séquence des paramètres obligatoires et optionnels. Les paramètres doivent être entourés par des parenthèses à gauche et à droite. Ce message peut s'afficher si le nombre de paramètres et leurs virgules de séparation ne sont pas valides. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" #2 est la position du paramètre, par exemple "3" Comme dans "INSTRING-Paramètre incorrect 3"

Nombre	Message	Explication
674	#1- Paramètre manquant #2	La syntaxe d'une fonction mentionne la séquence des paramètres obligatoires et optionnels. Les paramètres doivent être contournés par des parenthèses à gauche et à droite. Ce message peut s'afficher si le nombre de paramètres et leurs virgules de séparation ne sont pas valides. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" #2 est la position du paramètre, par exemple "2" Comme dans "INSTRING-Paramètre manquant 2"
675	#1- Séparateur incorrect #2	La syntaxe d'une fonction mentionne la séquence des paramètres obligatoires et optionnels. Les paramètres doivent être contournés par des parenthèses à gauche et à droite. Ce message peut s'afficher si le nombre de paramètres et leurs virgules de séparation ne sont pas valides. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" #2 est la position du séparateur, par exemple "3" Comme dans "INSTRING-Séparateur incorrect 3"
676	---	
677	#1-Paramètre #2 doit être #3	La syntaxe d'une fonction mentionne la séquence des paramètres obligatoires et optionnels. Les paramètres doivent être d'un type de données spécifique. Ce message peut s'afficher si le nombre de paramètres et leurs virgules de séparation ne sont pas valides. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" #2 est la position du paramètre, par exemple "1" #3 est le type de données requis pour le paramètre, par exemple "Alphanumérique" Comme dans "INSTRING-Paramètre 1 Doit être Alphanumérique"
678	---	
679	#1-Encodage des chaînes de caractères différent	Les fonctions de chaîne de caractères peuvent être utilisées sur des chaînes de caractères, mais tous les paramètres de chaîne de caractères doivent être du même type. #1 est le nom de la fonction, par exemple "INSTRING" Comme dans "INSTRING-Encodage des chaînes de caractères différent"
680	#1-Trop de paramètres	La syntaxe d'une fonction mentionne le nombre de paramètres obligatoires et optionnels. Ce nombre ne peut pas être dépassé. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" Comme dans "INSTRING-Trop de paramètres"

Nombre	Message	Explication
681	#1-Paramètre hors de la parenthèse fermante	La syntaxe d'une fonction mentionne le nombre de paramètres obligatoires et optionnels. La fin d'appel de fonction est la parenthèse de fermeture; aucun paramètre n'est autorisé après ce point. Comparez la syntaxe de votre appel de fonction dans MetaMap avec le diagramme de syntaxe fourni par IKAN. #1 est le nom de la fonction, par exemple "INSTRING" Comme dans "INSTRING-Paramètre hors de la parenthèse fermante"
682	Le fichier de Lignes séquentiel contient un champ signé	Les types de données flottant, binaire, packé et zoné ne sont pas permis pour les Fichiers de Lignes séquentielles.
683	Le fichier Lignes séquentiel contient un champ Unicode	Les types de données National et PRN-national ne sont pas permis pour les Fichiers de Lignes séquentielles.
684	L'usage d'Unicode est uniquement supporté pour IBM	Les types de données Unicode ne sont supportés que pour les environnements Cible IBM (Module Unicode).
685	Longueur du Masque d'édition différente de la taille	Pour les Champs PRN: le masque d'édition doit être égal à la taille de Champ. CORRECT = TYPE PRN SIZE 9 DECIMAL 2 EDIT 'Z(6)9V9(2)' total of the mask = 9 (6+1+2) INCORRECT = TYPE PRN SIZE 9 DECIMAL 2 EDIT 'Z(7)9V9(2)' total of the mask = 10 (7+1+2)
686	#1 est un CCSID non valide (non déclaré)	Un CCSID ne peut être assigné qu'à un Champ A (ASCII), E (EBCDIC), ou U (UNICODE). Il pourrait y avoir une erreur dans la table de Dictionnaire UCD001.
687	Masque avec décimales; aucune décimale définie	Le masque d'édition contient des décimales (comme dans 'Z(7)9V9(2)') mais le Champ de date n'est pas défini comme contenant des décimales.
688	Décimales différentes de celles du Masque d'édition	Le masque d'édition contient un nombre de décimales (comme dans 'Z(7)9V9(2)') mais le Champ de date est défini comme contenant un nombre de décimales différent).
689	Taille de chaînes remplacées et à remplacer différente	La longueur de la chaîne de caractères à remplacer doit être égale à la longueur de la chaîne de caractères remplaçante. #1 est le nom de la fonction, par exemple "REPLACE-ALL" #2 est la position du paramètre, par exemple "Différente" Comme dans "REPLACE-ALL – Longueur à remplacer & Longueur remplacée Différente"
690	---	La longueur de la chaîne de caractères à remplacer doit être égale à la longueur de la chaîne de caractères remplaçante. #1 est le nom de la fonction, par exemple "REPLACE-ALL" #2 est la position du paramètre, par exemple "Longueur" Comme dans "REPLACE-ALL – Longueur à remplacer > Longueur de la chaîne de caractères"
691	Modèle sans fichier Cible	Un Modèle MetaMap doit contenir au moins un Fichier Cible.

Nombre	Message	Explication
692	#1-FUNCTION #2 inconnue	La table OBFCVT déclare toutes les fonctions définies par l'Utilisateur. Crée le lien entre le nom externe tel qu'utilisé dans MetaMap et le nom interne utilisé par le Générateur MetaSuite. Toutes les fonctions définies par l'Utilisateur, y comprises les fonctions certifiées par IKAN, doivent être listées dans la table OBFCVT. Le nom externe doit être utilisé dans MetaMap (Module d'obscurcissement). OBFCVT <pre>>* Function Name-----> OBx R M Parameters RANDOM-INIT 001 N 0 N</pre> MetaMap MY-TARGET-FIELD = RANDOM-INIT (123456789).
693	Numéro de fonction non valide	La table OBFCVT déclare toutes les fonctions définies par l'Utilisateur. Crée le lien entre le nom externe tel qu'utilisé dans MetaMap et le nom interne utilisé par le Générateur MetaSuite. Le nom de la table comporte deux parties: 1. les trois caractères obligatoire "OBP" 2. un numéro compris entre 001 et 999 qui doit être spécifié dans la colonne OBx de la table OBFCVT. Exemple: <pre>>* Function Name-----> OBx R M Parameters RANDOM-INIT 001 N 0 N</pre>
694	---	
695	La logique pour #1-FUNCTION #2 n'est pas trouvée	La table OBFCVT déclare toutes les fonctions définies par l'Utilisateur. Crée le lien entre le nom externe tel qu'utilisé dans MetaMap et le nom interne utilisé par le Générateur MetaSuite. Toutes les fonctions définies par l'Utilisateur, y comprises les fonctions certifiées par IKAN, doivent être listées dans la table OBFCVT. Les données et la logique d'une fonction sont implémentées comme une table de Dictionnaire (OBFnnn) qui doit exister dans le Dictionnaire MetaSuite. Cette table n'est pas trouvée (Module d'obscurcissement).
696	Pas assez de paramètres obligatoires	Pour chaque fonction, la table OBFCVT mentionne combien de paramètres obligatoires doivent être fournis au moment où la fonction est appelée. Ce message est retourné si le nombre de paramètres fournis dans MetaMap est inférieur au nombre déclaré dans la table OBFCVT (Module d'obscurcissement).

Nombre	Message	Explication
697	Trop de paramètres fournis	Pour chaque fonction, la table OBFCVT mentionne combien de paramètres obligatoires doivent être fournis au moment où la fonction est appelée ainsi que le type de données de chaque paramètre, y compris les paramètres optionnels. Vous devez fournir le nombre de paramètres correct tel que déclaré dans la table OBFCVT (Module d'obscurcissement). OBFCVT <pre>>* Function Name-----> OBx R M Parameters RANDOM-INIT 001 N 0 N only one parameter</pre> MetaMap MY-TARGET-FIELD = RANDOM-INIT (123456789, "this parameter is not valid !").
698	Trop de paramètres littéraux dans le Modèle	Le maximum de 999 littéraux dans un Modèle MetaMap est atteint (Module d'obscurcissement).
699	#1 utilisé comme fonction système et fonction utilisateur	
700	Aucun XPath absolu n'est disponible pour XML	Ce message s'applique aux Fichiers XML. Le Chemin d'Enregistrement est un Chemin relatif, mais le Chemin de Fichier, auquel il est relatif, n'est pas disponible.

Messages 701 à 711

Nombre	Message	Explication
701	Le chemin de fichier doit être un chemin absolu	
702	XPATH non disponible pour:	Le XPATH pour un Enregistrement doit être spécifié au niveau de l'Enregistrement si aucun XPATH n'est présent au niveau du Fichier.
703	Attribut pour le Chemin de Fichier : non supporté	Aucun XPATH spécifié pour le niveau d'Enregistrement. Le Champ au niveau le plus élevé est un attribut. Il doit être attaché à l'Enregistrement pour lequel aucun XPATH n'est spécifié. Solution: raccourcissez le XPATH d'un niveau de Fichier et ajoutez le niveau omis au XPATH de l' (des)Enregistrement(s).
704	Le champ #1 n'appartient pas à la matrice externe	L'usage de SYS-CURRENT-KEY et SYS-RANDOM-KEY dans une matrice est limité aux matrices externes.
705	#1 non permis dans une Procédure de fichier Initial	Ce mot clé ne peut pas être utilisé dans une Procédure de Fichier initiale. Par exemple: SYS-RANDOM-KEY non permis dans une Procédure de Fichier initiale
706	Nœud vide dans le XPATH de l'enregistrement Cible #1	Le XPATH de l'Enregistrement Cible n'est pas vide, mais certains de ses nœuds sont vides. Par exemple: si le XPATH est '/X/Y//Z', il y a un écart entre Y et Z. Ces types de XPATH sont non valides.

Nombre	Message	Explication
707	Expression de chaîne trop longue	L'expression de chaîne de caractères dépasse les limites permises. Par exemple: la chaîne de caractères dans une instruction DEBUG ne peut pas dépasser la taille de 42 caractères.
708	Mot clé HEADER non supporté dans ce contexte	
709	END-CASE sans CASE	L'instruction END-CASE est rencontrée sans l'instruction correspondante CASE.
710	END-FOR sans FOR	L'instruction END-FOR est rencontrée sans l'instruction correspondante FOR.
711	Séquence FOR/END-FOR non valide	L'instruction END-FOR se trouve à un endroit non valide.

A

- ADD 34
- ADD TABLE 95
- Appeler le Générateur en Batch 149
- Appeler le Générateur en mode batch
 - Exemple 150
 - Préparation 149
 - Utiliser MSBGEN.EXE 149
- Assistant Commandes 21
 - Options de CHANGE 24
 - Options de TABLE 33
- Autres fonctions 84
 - Paramètres initiaux du Générateur 84
 - Parcourir les dossiers 85

B

- Batch
 - Appeler le Générateur 149
- Boucles FOR-NEXT 107
- BUILD 25

C

- CALL Table 108
- CHANGE DEFAULT BUILD 89
- CHANGE DEFAULT CHARACTER-CCSID 94
- CHANGE DEFAULT DATE 89
- CHANGE DEFAULT DECIMAL 90
- CHANGE DEFAULT DYNAMIC 94
- CHANGE DEFAULT EXEC 91
- CHANGE DEFAULT INTERPUNCTION 91
- CHANGE DEFAULT NULL 92
- CHANGE DEFAULT NULLABLE 92
- CHANGE DEFAULT PAGE 93
- CHANGE DEFAULT QUOTE 93
- CHANGE DEFAULT SQL DIALECT 94
- CHANGE DEFAULT SQL-QUOTE 31, 94
- CHANGE DEFAULT UNICODE-CCSID 94
- CHARACTER-CCSID 32
- Clé de licence 9
- Codes de retour du Générateur 152
- Commandes
 - Langage d'installation 88
 - MTL 105
- Commandes du langage d'installation 88

- Commandes MTL 105
 - Boucles FOR-NEXT 107
 - CALL Table 108
 - Conventions de notation 106
 - DUMP 109
 - FREEZE 109
 - GOTO 110
 - NESTED IF 111
 - REMARK 112
 - SET 112
 - SKIP 113
 - Structures EVALUATE 114
 - Structures IF 115
 - Table EXIT 109
 - TRACE 116
 - Traitement des messages d'erreur 117
 - UNFREEZE 109
 - UNSET 120

- COPY 35
- COPY TABLE 95
- Créer un Dictionnaire/ Saisir la Clé de licence 6

D

- DATE BREAK YEAR 26
- DATE FORMAT 25
- DECIMAL 26
- DELETE 35
- DELETE TABLE 96
- Dictionnaire
 - Créer 6
- Dossiers par défaut 85
- DUMP 109
- DYNAMIC 31

E

- Écran de génération 15
- Écran des Options du Dictionnaire 12
- Écrans du Generator Manager
 - Générer 15
 - Options du Dictionnaire 12
- Écrans Generator Manager
 - Créer un Dictionnaire/ Saisir la Clé de licence 6
 - Maintenance des tables techniques 54
- EXEC 27
- EXPORT 35
- Expressions

MTL 101
Expressions MTL 101

F

Fichiers MDL 37
 Analyser la Source XML 45
 Cible MDL en XML 42
 Générer - MDL 38
 MDL Cible délimité 40
 Modifier 39
Fichiers MSM 47
Fichiers MXL 49
 Générer 50
 Générer plusieurs Fichiers MXL en mode batch 52
Fonctions
 MTL 102
Fonctions MTL 102
FREEZE 109

G

Generator Manager
 Autres fonctions 84
Générer
 Générer les fichiers MDL 37
 Implémenter les instructions MIL 18
 Les Fichiers MSM 47
 Les Fichiers MXL 49
GOTO 110

I

Implémenter les instructions MIL 18
 Ajouter les instructions MIL 21
 Assistant Commandes 21
 Copier des Fichiers MIL existants 20
IMPORT 36
Instructions MIL 18
INTERPUNCTION 27

L

Langage d'exécution 131
LANGUAGE D'EXPORT 32
Langage d'installation
 ADD TABLE 95
 CHANGE DEFAULT BUILD 89
 CHANGE DEFAULT CHARACTER-CCSID 94
 CHANGE DEFAULT DATE 89
 CHANGE DEFAULT DECIMAL 90
 CHANGE DEFAULT DYNAMIC 94
 CHANGE DEFAULT EXEC 91
 CHANGE DEFAULT INTERPUNCTION 91
 CHANGE DEFAULT NULL 92
 CHANGE DEFAULT NULLABLE 92
 CHANGE DEFAULT PAGE 93

CHANGE DEFAULT QUOTE 93
CHANGE DEFAULT SQL DIALECT 94
CHANGE DEFAULT SQL-QUOTE 94
CHANGE DEFAULT UNICODE-CCSID 94
COPY TABLE 95
DELETE TABLE 96
LIST DEFAULT 96
LIST TABLE 96
LIST VERSION 97
NEW 89
REMARKS 98

Langage d'installation MetaSuite 18

Langage de modèle
 Commandes MTL 105
 Expressions MTL 101
 Fonctions MTL 102
 Variables du Générateur 104
 Variables MTL 99
 Variables MTL liées à l'Enregistrement Cible 126
 Variables MTL liées à l'Enregistrement Source 124
 Variables MTL liées au Champ 127
 Variables MTL liées au Fichier Cible 125
 Variables MTL spécifiques au Fichier Source 123

Langage de Modèle - Référence 99

LIST 33
LIST DEFAULT 96
LIST TABLE 96
LIST VERSION 97

M

Maintenance des tables techniques 54
 Tables MGL 58
Messages d'erreur de syntaxe du Générateur 155
Messages d'erreur du Dictionnaire 156
Messages du Générateur 152, 157
 Codes de retour du Générateur 152
 Messages d'erreur de syntaxe du Générateur 155
 Messages d'erreur du Dictionnaire 156
Modules de chaînes de caractères 139
MSBGEN 149

N

NESTED IF 111
NEW 89
NULL 28
NULLABLE 28

O

Options de CHANGE 24
 BUILD 25
 CHANGE DEFAULT SQL-QUOTE 31
 CHARACTER-CCSID 32
 DATE BREAK YEAR 26
 DATE FORMAT 25

DECIMAL 26
 DYNAMIC 31
 EXEC 27
 GUILLEMETS 29
 INTERPUNCTION 27
 LANGAGE D'EXPORT 32
 NULL 28
 NULLABLE 28
 PAGE 29
 SQL 30
 UNICODE-CCSID 32
 Options de TABLE 33
 ADD 34
 COPY 35
 DELETE 35
 EXPORT 35
 IMPORT 36
 LIST 33
 Options du Dictionnaire 12

P

PAGE 29
 Paramètres initiaux (Générateur) 84
 Paramètres initiaux du Générateur 84

Q

QUOTE 29

R

réer un Dictionnaire 6
 REMARK 112
 REMARKS 98

S

Saisir une Clé de licence 9
 Se connecter 3
 SET 112
 SKIP 113
 SQL 30
 Structures EVALUATE 114
 Structures IF 115

T

Table EXIT 109
 Tables de Code-Contrôle 65
 Code-Contrôle Cible 71
 Code-Contrôle Source 67
 Format de table standard 67
 Tables de Code-Contrôle de l'Utilisateur 73, 74
 Tables MGL
 Code-Contrôle Cible 71

Code-Contrôle Source 67
 Format de table de code-contrôle standard 67
 Maintenance des tables techniques 59
 Tables de code 74
 Tables de code COBOL personnalisées 76
 Tables de Code-Contrôle 65
 Tables de Code-Contrôle de l'Utilisateur 73, 74
 Tables de Code-Contrôle personnalisées 76
 Tables MRL
 Maintenance des tables techniques MRL 79
 TRACE 116
 Traitement des messages d'erreur 117

U

UNFREEZE 109
 UNICODE-CCSID 32

V

Variables
 Générateur 104
 MTL 99
 Liées à l'Enregistrement Cible 126
 Liées au Champ 127
 Liées au Fichier Cible 125
 Système 121
 Variables liées à l'Enregistrement 124
 Variables spécifiques au Fichier Source 123
 Variables du Générateur 104
 Variables MTL 99
 Variables MTL liées à l'Enregistrement 124
 Variables MTL liées à l'Enregistrement Cible 126
 Variables MTL liées au Champ 127
 Variables MTL liées au Fichier Cible 125
 Variables MTL spécifiques au Fichier Source 123
 Variables Système MTL 121
 MTL 121