

IBM Planning Analytics  
Version 2 Edition 0

*TM1 - Guide de référence*



**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «[Mentions légales](#)», à la page 433.

**Informations sur le produit**

Ce document s'applique à IBM Planning Analytics version 2.0 et peut également s'appliquer aux éditions ultérieures.

Licensed Materials - Property of IBM. Éléments sous licence - Propriété d'IBM.

Dernière mise à jour : 2019-11-13

© **Copyright International Business Machines Corporation 2007, 2019.**

---

# Table des matières

<b>Introduction.....</b>	<b>XV</b>
<b>Chapitre 1. Fenêtres et boîtes de dialogue.....</b>	<b>1</b>
Boîte de dialogue Propriétés du bouton d'action.....	1
Onglet Processus.....	2
Onglet Feuille de calcul.....	3
Onglet Apparence.....	4
Boîte de dialogue Options avancées.....	5
Grille de mappage avancé.....	5
Editeur d'attributs.....	7
Menu Fichier.....	7
Menu Edition.....	7
Options de mise en forme.....	8
Fenêtre Journal d'audit.....	9
Panneau de requêtes.....	9
Panneau Résultats.....	12
Fenêtre Détail du journal d'audit.....	13
Barre d'outils Détails.....	13
Grille Détails.....	14
Assistant Définition de tâche.....	14
Ecran 1 (Etape 1).....	14
Ecran 2 (Etape 2).....	15
Fenêtre Clients/Groupes.....	15
Menu Sécurité.....	16
Menu Clients.....	16
Menu Groupes.....	16
Grille Clients/Groupes.....	17
Boîte de dialogue Centre de messagerie clients.....	18
Boîte de dialogue Créer une dimension.....	18
Boîte de dialogue Créer un objet de réplication de serveur.....	19
Boîte de dialogue Créer un cube.....	19
Boîte de dialogue Optimiseur de cube.....	20
Boîte de dialogue Propriétés de cube.....	21
Cube Viewer.....	21
Menu Fichier.....	22
Menu Edition.....	22
Menu Affichage.....	23
Menu Options.....	23
Boîte de dialogue Supprimer les sous-ensembles nommés.....	24
Boîte de dialogue Supprimer les vues nommées.....	25
Editeur de dimension.....	25
Menu Dimension.....	25
Menu Edition.....	25
Menu Affichage.....	28
Boîte de dialogue Insertion d'élément de dimension.....	29
Boîte de dialogue Ordre des éléments de dimension.....	29
Boîte de dialogue Propriétés d'élément de dimension.....	30
Détailler .....	31
Boîte de dialogue Editer la formule.....	31
Boîte de dialogue Editer la référence au cube.....	32

Boîte de dialogue Filtrer les éléments par valeur d'attribut.....	32
Boîte de dialogue Filtrer les éléments par niveau.....	32
Boîte de dialogue Filtrer un sous-ensemble.....	33
Boîte de dialogue Filtrer une vue.....	34
Boîte de dialogue Choisir une vue (Navigateur de feuille de calcul).....	36
Menu Navigateur de feuille de calcul.....	36
Fenêtre Journal des messages.....	37
Menu Fichier.....	37
Menu Edition.....	38
Menu Aide.....	38
Boîte de dialogue Nouvel attribut.....	38
Boîte de dialogue Ouvrir un sous-ensemble.....	38
Boîte de dialogue Ouvrir une vue.....	39
Assistant d'impression de rapports.....	39
Tous les écrans.....	39
Ecran 1 de 3.....	39
Ecran 2 de 3.....	40
Ecran 3 de 3.....	41
Boîte de dialogue Options de processus.....	43
Boîte de dialogue Répliquer un cube.....	44
Informations sur le cube.....	44
Informations sur les règles.....	45
Informations sur les dimensions.....	45
Editeur de règles.....	46
Menu Fichier.....	46
Menu Edition.....	47
Menu Affichage.....	48
Menu Insertion.....	49
Menu Outils.....	49
Boîte de dialogue Enregistrer le sous-ensemble.....	49
Boîte de dialogue Enregistrer la vue.....	50
Boîte de dialogue Enregistrer la vue (Navigateur de feuille de calcul).....	50
Boîte de dialogue Affectations de sécurité.....	51
Grille Affectations.....	51
Privilèges d'accès.....	51
Sélectionner une dimension.....	55
Boîte de dialogue Sélectionner un cube.....	55
Boîte de dialogue Sélectionner un cube pour les règles.....	55
Boîte de dialogue Sélectionner une dimension.....	55
Boîte de dialogue Sélectionner un élément.....	55
Explorateur de serveur (Fenêtre principale).....	55
Menu Fichier.....	56
Menu dynamique.....	56
Menu Edition.....	67
Menu Affichage.....	68
Editeur de sous-ensemble.....	68
Menu Sous-ensemble.....	69
Menu Edition.....	69
Menu Affichage.....	72
Menu Outils.....	73
Boîte de dialogue Alias.....	74
Boîte de dialogue Options TM1.....	74
Paramètres de connexion.....	74
Serveur local.....	74
Protocole TSL (Transport Layer Security) du serveur Admin.....	74
Boîte de dialogue Requête du journal des transactions.....	75
Boîte de dialogue Résultats de la requête du journal des transactions.....	76
Editeur TurboIntegrator.....	77

Menu Fichier.....	77
Menu Edition.....	78
Onglet Source de données.....	78
Aperçu de grille.....	92
Onglet Variables.....	92
Onglet Mappages.....	94
Onglet Expert.....	98
Onglet Planning.....	100
Fenêtre Extraction de vue.....	100
Boîte de dialogue Styles de vue.....	101

## **Chapitre 2. Fonctions de règles..... 103**

Opérateurs arithmétiques dans les règles TM1.....	103
Opérateurs de comparaison dans les règles TM1.....	103
Opérateurs logiques dans les règles TM1.....	103
Fonctions de règles d'attribut.....	104
ATTRN.....	104
ATTRS.....	105
CubeATTRN.....	106
CubeATTRS.....	106
DimensionATTRN.....	106
DimensionATTRS.....	107
ElementAttrN.....	107
ElementAttrS.....	108
Fonctions de règles de calcul de consolidation.....	109
ConsolidatedAvg.....	109
ConsolidateChildren.....	110
ConsolidatedCount.....	112
ConsolidatedCountUnique.....	113
ConsolidatedMax.....	113
ConsolidatedMin.....	114
Fonctions de règles pour données de cube.....	115
CellValueN.....	115
CellValueS.....	116
DB.....	117
ISLEAF.....	117
ISUNDEFINEDCELLVALUE.....	118
UNDEF.....	118
UNDEFINEDCELLVALUE.....	119
UNDEFVALS.....	119
Fonctions de règles de date et heure.....	121
DATE.....	121
DATES.....	121
DAY.....	122
DAYNO.....	122
MONTH.....	123
NOW.....	123
TIME.....	123
TIMST.....	123
TIMVL.....	125
TODAY.....	127
YEAR.....	128
Fonctions de règles d'informations de dimensions.....	128
DIMIX.....	129
DIMNM.....	129
DIMSIZ.....	129
DNEXT.....	130

DNLEV.....	130
DTYPE .....	131
TABDIM.....	131
Fonctions de règles d'informations d'éléments.....	132
ELCOMP .....	132
ELCOMPN.....	132
ElementComponent .....	133
ElementComponentCount.....	133
ElementCount .....	134
ElementFirst.....	134
ElementIndex.....	135
ElementIsAncestor.....	135
ElementIsComponent.....	136
ElementIsParent .....	136
ElementLevel.....	137
ElementName.....	138
ElementNext.....	138
ElementParent.....	139
ElementParentCount.....	139
ElementType .....	140
ElementWeight .....	140
ELISANC.....	141
ELISCOMP .....	141
ELISPAR .....	142
ELLEV.....	142
ELPAR.....	143
ELPARN.....	143
ELWEIGHT .....	144
LevelCount.....	144
Fonctions de règles financières.....	145
VC.....	145
PAYMT .....	146
PV.....	146
Fonctions de règles de hiérarchie.....	147
Hierarchy.....	147
HierarchyCount.....	147
HierarchyIndex.....	147
HierarchyN.....	148
Fonctions de règles logiques.....	148
CONTINUE.....	148
IF.....	149
STET.....	149
Fonctions de règles mathématiques.....	150
ABS.....	150
ACOS.....	150
ASIN.....	150
ATAN.....	151
COS.....	151
EXP.....	151
ENT.....	152
ISUND.....	152
LN.....	152
LOG.....	153
MAX.....	153
MIN .....	153
MOD.....	154
RAND.....	154
ARRONDI.....	154

ROUNDUP.....	155
SIGN.....	156
SIN.....	156
SQRT.....	156
TAN.....	157
Fonctions de règles de texte.....	157
CAPIT.....	157
CHAR.....	157
CODE.....	158
DELET.....	158
FILL.....	159
INSRT.....	159
LONG.....	159
LOWER.....	160
NUMBR.....	160
SCAN.....	160
STR.....	161
SUBST.....	163
TRIM.....	163
UPPER.....	164
Fonctions de règles diverses.....	164
FEEDERS.....	164
FEEDSTRINGS.....	164
SKIPCHECK.....	165

### **Chapitre 3. Fonctions de macros.....167**

Accès aux fonctions macro de Microsoft Excel 2010 et versions ultérieures.....	167
Accès aux fonctions de macros dans les modules VBA.....	167
D_PICK.....	167
D_FSAVE.....	168
D_SAVE.....	169
DBProportionalSpread.....	169
E_PICK.....	170
I_EXPORT.....	171
I_NAMES.....	172
I_PROCESS.....	172
M_CLEAR.....	173
N_CONNECT.....	173
N_DISCONNECT.....	173
OPTGET.....	174
OPTSET.....	174
PublishSubset.....	175
PublishView.....	176
QUDEFINE.....	176
QUDEFINEEX.....	178
QUEXPORT.....	180
QULOOP.....	181
QUSUBSET.....	181
R_SAVE.....	182
SUBDEFINE.....	182
SUBPICK.....	183
T_CLEAR.....	184
T_CREATE.....	184
T_CREATE16.....	185
T_PICK.....	186
T_SAVE.....	186
TM1RECALC.....	186

TM1RECALC1.....	187
VUSLICE.....	187
W_DBSENABLE.....	187

## **Chapitre 4. Fonctions de feuilles de calcul..... 189**

DBR.....	189
DBRA.....	190
DBRW.....	191
DBS.....	192
DBSA.....	192
DBSS.....	193
DBSW.....	194
DFRST.....	195
DIMIX.....	195
DIMNM.....	196
DIMSIZ.....	196
DNEXT.....	196
DNLEV.....	197
DTYPE.....	198
ELCOMP.....	198
ELCOMPN.....	199
ELISCOMP.....	199
ELISPAR.....	200
ELLEV.....	200
ELPAR.....	201
ELPARN.....	202
ELSLEN.....	202
ELWEIGHT.....	203
SUBNM.....	203
SUBSIZ.....	204
TABDIM.....	205
TM1ELLIST.....	205
TM1GLOBALSANDBOX.....	208
TM1INFO.....	208
TM1PRIMARYDBNAME.....	209
TM1RptEIsConsolidated.....	209
TM1RptEIsExpanded.....	210
TM1RptELev.....	210
TM1RptFilter.....	210
TM1RptRow.....	211
TM1RptTitle.....	213
TM1RptView.....	214
TM1User.....	214
VIEW.....	215

## **Chapitre 5. fonctions TurboIntegrator..... 217**

Mots réservés de TurboIntegrator.....	217
Fonctions TurboIntegrator pour ASCII et texte.....	217
ASCIIDelete.....	218
ASCIIOutput.....	218
NumberToString.....	219
NumberToStringEx.....	220
SetInputCharacterSet.....	220
SetOutputCharacterSet.....	223
SetOutputEscapeDoubleQuote.....	223
StringToNumber.....	224
StringToNumberEx.....	224



TextOutput.....	225
Fonctions TurboIntegrator de manipulation d'attributs.....	226
ATTRNL.....	226
ATTRSL.....	227
AttrDelete.....	228
AttrInsert.....	229
AttrPutN.....	229
AttrPutS.....	230
ChoreAttrDelete.....	231
ChoreAttrInsert.....	231
ChoreAttrN.....	232
ChoreAttrNL.....	232
ChoreAttrPutN.....	233
ChoreAttrPutS.....	234
ChoreAttrS.....	235
ChoreAttrSL.....	235
CreateHierarchyByAttribute.....	236
CubeAttrDelete.....	237
CubeAttrInsert.....	237
CubeAttrPutN.....	238
CubeAttrPutS.....	239
CubeATTRNL.....	240
CubeATTRSL.....	241
DimensionAttrDelete.....	242
DimensionAttrInsert.....	242
DimensionAttrPutN.....	243
DimensionAttrPutS.....	244
DimensionATTRNL.....	244
DimensionATTRSL.....	245
ElementATTRNL.....	246
ElementATTRSL.....	247
ElementAttrPutN.....	248
ElementAttrPutS.....	249
ElementAttrInsert.....	250
ElementAttrDelete.....	251
HierarchyAttrPutN.....	251
HierarchyAttrPutS.....	252
HierarchyATTRN.....	253
HierarchyATTRS.....	253
HierarchyATTRNL.....	254
HierarchyATTRSL.....	255
HierarchySubsetATTRS.....	256
HierarchySubsetATTRN.....	256
HierarchySubsetATTRSL.....	257
HierarchySubsetATTRNL.....	258
HierarchySubsetAttrPutS.....	259
HierarchySubsetAttrPutN.....	260
HierarchySubsetAttrInsert.....	261
HierarchySubsetAttrDelete.....	262
ProcessAttrDelete.....	262
ProcessAttrInsert.....	263
ProcessAttrN.....	263
ProcessAttrNL.....	264
ProcessAttrPutN.....	265
ProcessAttrPutS.....	266
ProcessAttrS.....	267
ProcessAttrSL.....	267
SubsetATTRS.....	268

SubsetATTRN.....	269
SubsetATTRSL.....	269
SubsetATTRNL.....	270
SubsetAttrPutS.....	271
SubsetAttrPutN.....	272
SubsetAttrInsert.....	273
SubsetAttrDelete.....	274
ViewAttrDelete.....	274
ViewAttrInsert.....	275
ViewAttrN.....	275
ViewAttrNL.....	276
ViewAttrPutN.....	277
ViewAttrPutS.....	278
ViewAttrS.....	279
ViewAttrSL.....	279
Fonctions TurboIntegrator de gestion de tâches.....	280
ChoreError.....	281
ChoreQuit.....	281
ChoreRollback.....	281
SetChoreVerboseMessages.....	281
Fonctions TurboIntegrator de manipulation de cubes.....	282
AddCubeDependency.....	282
CellGetN.....	283
CellGetS.....	284
CellIncrementN.....	285
CellIsUpdateable.....	285
CellPutN.....	286
CellPutProportionalSpread.....	287
CellPutS.....	287
CubeClearData.....	288
CubeCreate.....	289
CubeDestroy.....	289
CubeDimensionCountGet.....	289
CubeExists.....	290
CubeGetLogChanges.....	290
CubeSaveData.....	291
CubeSetConnParams.....	292
CubeSetLogChanges.....	292
CubeTimeLastUpdated.....	293
CubeUnload.....	293
Fonctions TurboIntegrator de réservation de données.....	294
CubeDataReservationAcquire.....	294
CubeDataReservationRelease.....	295
CubeDataReservationReleaseAll.....	295
CubeDataReservationGet.....	296
CubeDataReservationGetConflicts.....	298
CubeDRAcquire.....	299
CubeDRRelease.....	300
CubeDRReleaseAll.....	300
CubeDRGet.....	301
CubeDRGetConflicts.....	303
Fonctions TurboIntegrator de date et d'heure.....	304
FormatDate.....	304
NewDateFormatter.....	304
ParseDate.....	305
Fonctions TurboIntegrator de manipulation de dimensions.....	306
DimensionCreate.....	306
DimensionDeleteAllElements.....	306

DimensionDeleteElements.....	307
DimensionDestroy.....	307
DimensionElementComponentAdd.....	308
DimensionElementComponentAddDirect.....	308
DimensionElementComponentDelete.....	309
DimensionElementComponentDeleteDirect.....	309
DimensionElementDelete.....	310
DimensionElementDeleteDirect.....	311
DimensionElementExists.....	311
DimensionElementInsert.....	312
DimensionElementInsertDirect.....	313
DimensionElementPrincipalName.....	314
DimensionExists.....	314
DimensionHierarchyCreate.....	315
DimensionSortOrder.....	315
DimensionTimeLastUpdated.....	317
DimensionTopElementInsert.....	317
DimensionTopElementInsertDirect.....	318
DimensionUpdateDirect.....	319
Fonctions TurboIntegrator de manipulation des hiérarchies.....	319
HierarchyContainsAllLeaves.....	319
HierarchyCreate.....	320
HierarchyDeleteAllElements.....	320
HierarchyDeleteElements.....	321
HierarchyDestroy.....	321
HierarchyElementComponentAdd.....	321
HierarchyElementComponentAddDirect.....	322
HierarchyElementComponentDelete.....	323
HierarchyElementComponentDeleteDirect.....	323
HierarchyElementDelete.....	324
HierarchyElementDeleteDirect.....	325
HierarchyElementExists.....	326
HierarchyElementInsert.....	326
HierarchyElementInsertDirect.....	327
HierarchyElementPrincipalName.....	328
HierarchyExists.....	329
HierarchyHasOrphanedLeaves.....	329
HierarchySortOrder.....	330
HierarchyTimeLastUpdated.....	331
HierarchyTopElementInsert.....	331
HierarchyTopElementInsertDirect.....	332
HierarchyUpdateDirect.....	333
Fonctions TurboIntegrator ODBC.....	333
ODBCClose.....	334
ODBCOpen.....	334
ODBCOPENEx.....	334
ODBCOutput.....	335
SetODBCUnicodeInterface.....	336
Fonctions TurboIntegrator de contrôle de processus.....	336
ExecuteCommand.....	336
ExecuteProcess.....	337
GetProcessErrorFileDirectory.....	338
GetProcessErrorFilename.....	338
GetProcessName.....	339
If.....	339
ItemReject.....	340
ItemSkip.....	340
ProcessBreak.....	341

ProcessError.....	341
ProcessExists.....	341
ProcessQuit.....	342
ProcessRollback.....	342
RunProcess.....	342
Synchronized.....	343
While.....	344
Fonctions TurboIntegrator de gestion de règles.....	345
CubeProcessFeeders.....	345
CubeRuleAppend.....	345
CubeRuleDestroy.....	346
DeleteAllPersistentFeeders.....	347
ForceSkipCheck.....	347
RuleLoadFromFile.....	347
Fonctions de bac à sable.....	348
GetUseActiveSandboxProperty.....	348
ServerActiveSandboxGet.....	349
ServerActiveSandboxSet.....	349
ServerSandboxClone.....	350
ServerSandboxCreate.....	350
ServerSandboxesDelete.....	351
ServerSandboxDiscardAllChanges.....	354
ServerSandboxMerge.....	355
ServerSandboxExists.....	356
ServerSandboxGet.....	356
ServerSandboxListCountGet.....	357
SetUseActiveSandboxProperty.....	358
Fonctions TurboIntegrator de sécurité.....	358
AddClient.....	359
AddGroup.....	359
AssignClientToGroup.....	359
AssignClientPassword.....	360
AssociateCAMIDToGroup.....	360
CellSecurityCubeCreate.....	361
CellSecurityCubeDestroy.....	361
DeleteClient.....	362
DeleteGroup.....	362
ElementSecurityGet.....	363
ElementSecurityPut.....	363
HierarchyElementSecurityGet.....	364
HierarchyElementSecurityPut.....	364
RemoveCAMIDAssociation.....	365
RemoveCAMIDAssociationFromGroup.....	366
RemoveClientFromGroup.....	366
SetHierarchyGroupsSecurity.....	367
SetHierarchyElementGroupsSecurity.....	367
SetDimensionGroupsSecurity.....	368
SetElementGroupsSecurity.....	368
SecurityOverlayGlobalLockCell.....	369
SecurityOverlayCreateGlobalDefault.....	370
SecurityOverlayDestroyGlobalDefault.....	370
SecurityOverlayGlobalLockNode.....	371
SecurityRefresh.....	372
Fonctions TurboIntegrator de manipulation de serveur.....	372
BatchUpdateFinish.....	372
BatchUpdateFinishWait.....	373
BatchUpdateStart.....	374
DisableBulkLoadMode.....	374

EnableBulkLoadMode.....	375
RefreshMdxHierarchy.....	375
SaveDataAll.....	376
ServerShutdown.....	377
Fonctions TurboIntegrator de manipulation de sous-ensembles.....	377
HierarchySubsetAliasSet.....	378
HierarchySubsetCreate.....	378
HierarchySubsetDeleteAllElements.....	379
HierarchySubsetDestroy.....	379
HierarchySubsetElementExists.....	380
HierarchySubsetElementDelete.....	380
HierarchySubsetElementGetIndex.....	381
HierarchySubsetElementInsert.....	382
HierarchySubsetExists.....	382
HierarchySubsetGetSize.....	383
HierarchySubsetGetElementName.....	383
HierarchySubsetIsAllSet.....	384
HierarchySubsetMDXGet.....	384
HierarchySubsetMDXSet.....	384
SubsetAliasSet.....	385
SubsetCreate.....	386
SubsetCreateByMDX.....	387
SubsetDeleteAllElements.....	388
SubsetDestroy.....	389
SubsetElementDelete.....	389
SubsetElementExists.....	390
SubsetElementGetIndex.....	390
SubsetElementInsert.....	391
SubsetExists.....	391
SubsetExpandAboveSet.....	392
SubsetFormatStyleSet.....	393
SubsetGetElementName.....	393
SubsetGetSize.....	394
SubsetIsAllSet.....	394
SubsetMDXGet.....	394
SubsetMDXSet.....	395
Fonctions TurboIntegrator de manipulation de vues.....	395
PublishView.....	396
DisableMTQViewConstruct.....	396
EnableMTQViewConstruct.....	397
ViewColumnDimensionSet.....	398
ViewColumnSuppressZeroesSet.....	398
ViewConstruct.....	399
ViewCreate.....	399
ViewCreateByMDX.....	400
ViewDestroy.....	401
ViewExists.....	402
ViewExtractFilterByTitlesSet.....	402
ViewExtractSkipCalcsSet.....	403
ViewExtractSkipConsolidatedStringsSet.....	404
ViewExtractSkipRuleValuesSet.....	405
ViewExtractSkipZeroesSet.....	405
ViewMDXSet.....	406
ViewMDXGet.....	407
ViewRowDimensionSet.....	407
ViewRowSuppressZeroesSet.....	408
ViewSubsetAssign.....	409
ViewSuppressZeroesSet.....	409

ViewTitleDimensionSet.....	410
ViewTitleElementSet.....	410
ViewZeroOut.....	411
Fonctions TurboIntegrator diverses.....	411
AddInfoCubeRestriction.....	411
ExecuteJavaN.....	412
ExecuteJavaS.....	413
Expand.....	414
FileExists.....	415
LogOutput.....	415
TM1User.....	416
WildcardFileSearch.....	416
<b>Chapitre 6. Variables TurboIntegrator.....</b>	<b>419</b>
Variables locales TurboIntegrator.....	419
DatasourceASCIIDecimalSeparator.....	419
DatasourceASCIIDelimiter.....	419
DatasourceASCIIHeaderRecords.....	420
DatasourceASCIIQuoteCharacter.....	420
DatasourceASCIIThousandSeparator.....	421
DatasourceCubeview.....	421
DatasourceDimensionSubset.....	421
DatasourceNameForServer.....	422
DatasourceNameForClient.....	422
DataSourceODBOCatalog.....	422
DataSourceODBOConnectionString.....	423
DataSourceODBOCubeName.....	423
DataSourceODBOHierarchyName.....	423
DataSourceODBOLocation.....	424
DataSourceODBOProvider.....	424
DataSourceODBOSAPClientID.....	424
DataSourceODBOSAPClientLanguage.....	425
DatasourcePassword.....	425
DatasourceQuery.....	425
DataSourceSAPUsingRoleAuths.....	425
DataSourceSAPUsingTexts.....	426
DatasourceType.....	426
DatasourceUsername.....	426
MinorErrorLogMax.....	427
NValue.....	427
OnMinorErrorDoItemSkip.....	428
SValue.....	428
Fichier TM1ProcessError.log.....	428
Value_Is_String.....	429
Variables globales TurboIntegrator.....	429
NumericGlobalVariable('NomVariable');.....	430
StringGlobalVariable('NomVariable');.....	430
Variables globales implicites.....	430
DataMinorErrorCount.....	430
MetadataMinorErrorCount.....	431
ProcessReturnCode.....	431
PrologMinorErrorCount.....	432
Variables utilisateur TurboIntegrator.....	432
<b>Mentions légales.....</b>	<b>433</b>
<b>Index.....</b>	<b>437</b>

# Introduction

---

Ce document est destiné à être utilisé avec IBM® Planning Analytics.

Ce document est un ensemble de documents de référence qui décrit les fonctions logicielles, variables et autres éléments de programmation d'IBM Planning Analytics.

Planning Analytics fournit des solutions logicielles pour la gestion et la surveillance permanentes des performances financières et opérationnelles dans l'entreprise.

## Clause de décharge relative aux exemples

La société Vacances et Aventure, Ventes GO, et toutes les variantes du nom Vacances et Aventure, ainsi que l'Exemple de planification, décrivent des opérations métier fictives. Celles-ci contiennent des données qui servent de modèle à IBM et à ses clients pour développer des applications modèles. Les exemples de données contenus dans ces enregistrements fictifs sont utilisés pour représenter des transactions de vente, la distribution de produits, des données financières et des ressources humaines. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite. D'autres fichiers d'exemple peuvent contenir des données fictives générées manuellement ou par une machine, des données factuelles compilées à partir de sources académiques ou publiques, ou des données utilisées avec l'autorisation du détenteur des droits d'auteur, à utiliser comme échantillon de données pour développer des exemples d'application. Les noms de produit référencés peuvent être les marques de leurs propriétaires respectifs. Toute reproduction sans autorisation est interdite.

## Fonctions d'accessibilité

Les fonctions d'accessibilité aident les utilisateurs qui souffrent d'un handicap physique, tel qu'une mobilité limitée ou une vision faible, à utiliser les logiciels.

Ce produit ne prend actuellement en charge aucune fonction d'accessibilité destinée à aider les utilisateurs ayant une incapacité physique, telle qu'une mobilité ou une vision réduites, à utiliser ce produit.

## Instructions prospectives

La présente documentation décrit les fonctionnalités actuelles du produit. Elle peut contenir des références à des éléments qui ne sont pas disponibles actuellement. Cela n'implique aucune disponibilité ultérieure de ces éléments. De telles références ne constituent en aucun cas un engagement, une promesse ou une obligation légale de fournir un élément, un code ou une fonctionnalité. Le développement, la disponibilité et le calendrier de mise à disposition des fonctions demeurent à la seule discrétion d'IBM.

## Considérations de sécurité

Pour les considérations de sécurité d'IBM Planning Analytics, voir *Planning Analytics - Installation et configuration*. Vous trouverez des informations sur la gestion de l'authentification des utilisateurs et des groupes dans le chapitre *Gestion des utilisateurs et des groupes* du document *TM1 - Guide des opérations*.





---

# Chapitre 1. Fenêtres et boîtes de dialogue

Cette section décrit les principales fenêtres et boîtes de dialogue d'IBM Planning Analytics.

## Boîte de dialogue Propriétés du bouton d'action

---

Utilisez la boîte de dialogue Propriétés du bouton d'action pour ajouter des boutons d'action TM1 à une feuille de calcul. Vous pouvez configurer le bouton pour exécuter un processus et aller jusqu'à une feuille de calcul.

Des exemples et des instructions sur l'utilisation des boutons d'action dans les feuilles de calcul figurent dans le document *TM1 for Developers* dans l'[IBM Knowledge Center](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0) ([https://www.ibm.com/support/knowledgecenter/SSD29G\\_2.0.0](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0)).

### Serveur

Cette liste contient le nom de tous les serveurs TM1 actuellement disponibles sur votre réseau.

Sélectionnez le serveur sur lequel le processus ou la feuille de calcul cible du bouton d'action sont situés.

### Connecter

Ce bouton est disponible uniquement quand vous n'êtes pas connecté au serveur sélectionné dans la zone de liste des serveurs.

Cliquez sur ce bouton pour vous connecter au serveur que vous avez sélectionné dans la zone de liste des serveurs.

### Déconnecter

Ce bouton est disponible uniquement quand vous êtes connecté au serveur sélectionné dans la zone de liste des serveurs.

Cliquez sur ce bouton pour vous déconnecter du serveur que vous avez sélectionné dans la zone de liste des serveurs.

### Action

Sélectionnez l'action que le bouton d'action doit effectuer lorsque vous cliquez dessus.

- Exécuter un processus TurboIntegrator

Sélectionnez cette option pour configurer un bouton d'action qui exécute un processus. Lorsque vous sélectionnez cette option, l'onglet Processus est activé.

- Aller à une autre feuille de calcul

Sélectionnez cette option pour configurer un bouton d'action permettant d'accéder à une autre feuille de calcul. Lorsque vous sélectionnez cette option, l'onglet Feuille de calcul est activé.

- Exécuter un processus puis aller à une feuille de calcul

Sélectionnez cette option pour configurer un bouton d'action qui exécute un processus, puis accède à une autre feuille de calcul. Lorsque vous sélectionnez cette option, les onglets Processus et Feuilles de calcul sont activés.

- Calculer / Régénérer uniquement

Sélectionnez cette option pour recalculer ou régénérer sans exécuter de processus TI ou accéder à une autre feuille de calcul. Cette option peut être utile si vous désirez seulement mettre à jour la feuille en cours ou recharger la version d'origine d'un formulaire actif.

Vous pouvez également utiliser l'onglet **Calculer** pour sélectionner l'opération de calcul que TM1 doit effectuer *avant* l'exécution d'un processus TI ou la navigation vers une autre feuille de calcul.

## OK

Ferme la boîte de dialogue Propriétés du bouton d'action et insère un bouton d'action dans une feuille de calcul.

## Annuler

Ferme la boîte de dialogue Propriétés du bouton d'action sans insérer un bouton d'action.

## Onglet Processus

Utilisez l'onglet Processus pour configurer un bouton d'action pour exécuter un processus.

### Processus

Utilisez cette liste pour sélectionner le processus que vous souhaitez exécuter de l'une des manières suivantes :

- Pour sélectionner un processus disponible sur le serveur en cours, sélectionnez le nom du processus dans la liste.
- Pour extraire le nom du processus et les valeurs de paramètre de la feuille de calcul en cours, sélectionnez Extraire les informations de processus de la feuille de calcul.

### Options

Ouvre la boîte de dialogue Options du processus dans laquelle vous pouvez contrôler le comportement du bouton d'action *avant* et *après* l'exécution du processus.

Pour plus d'informations, voir [«Boîte de dialogue Options de processus»](#), à la page 43.

### Nom du processus

Cette option s'affiche uniquement quand vous sélectionnez Extraire les informations de processus de la feuille de calcul dans la liste Processus.


Entrez une référence Excel donnant le nom du processus à exécuter de l'une des manières suivantes.

- Utilisez le format ci-dessous pour référencer une seule cellule : =ColumnNameRowName. Par exemple : =A1.
- Si vous souhaitez référencer une plage nommée dans Excel, utilisez le format suivant : =NameOfRange
- Cliquez sur le bouton **Référence Excel** adjacent à la zone Nom du processus pour sélectionner une référence dans la feuille de calcul en cours.

### Paramètres

Entrez les valeurs des paramètres de processus, selon la façon dont vous avez sélectionné le nom du processus dans la liste Processus.

- Si vous avez sélectionné un processus dans la liste Processus, la grille Paramètres s'affiche avec une liste des paramètres pour le processus sélectionné. Vous pouvez entrer les valeurs de chaque paramètre directement dans la grille, ou utiliser une référence Excel extrayant dynamiquement une valeur de paramètre dans la feuille de calcul en cours.
- Si vous avez sélectionné l'option Extraire les informations de processus de la feuille de calcul dans la liste Processus, vous devez utiliser une référence Excel pour extraire les valeurs de paramètre de la feuille de calcul en cours. Vous pouvez entrer une référence à une seule cellule, une plage de cellules ou une plage nommée. Toute référence doit pointer sur le nombre de cellules approprié, en fonction des paramètres attendus par le processus.

Cliquez sur le bouton **Référence Excel**  pour sélectionner directement la cellule ou la plage de cellules dans la feuille de calcul.

Pour des exemples, reportez-vous à la documentation *TM1 for Developers*.

## Référence Excel

Crée une référence Excel extrayant dynamiquement le nom du processus ou la/les valeurs de paramètres dans la feuille de calcul en cours lorsque vous cliquez sur le bouton **Action**.

## Onglet Feuille de calcul

Utilisez l'onglet Feuille de calcul pour configurer un bouton d'action permettant d'aller à une autre feuille de calcul Excel.

### Chercher dans

Utilisez une des méthodes suivantes pour sélectionner une feuille de calcul :

- Applications TM1 - Sélectionnez cette option si vous souhaitez sélectionner une feuille de calcul dans l'arborescence des applications TM1.
- Fichiers - Sélectionnez cette option si vous souhaitez choisir une feuille de calcul sur votre ordinateur.

### Parcourir

Cliquez sur ce bouton pour sélectionner la feuille de calcul à laquelle vous souhaitez accéder.

- Si vous avez sélectionné l'option Applications TM1, une boîte de dialogue s'ouvre et vous permet de sélectionner une feuille de calcul dans l'arborescence des applications TM1.
- Si vous avez sélectionné l'option Fichiers la boîte de dialogue Ouvrir s'affiche. Vous pouvez parcourir les fichiers et en sélectionner un sur votre ordinateur.

### Classeur

Contient le chemin et le nom du classeur Excel auquel vous souhaitez aller. Vous pouvez entrer cette valeur de l'une des manières suivantes :

- Cliquez sur le bouton **Parcourir** situé à côté de l'option Chercher dans pour sélectionner un classeur dans l'arborescence des applications TM1 ou dans les fichiers qui se trouvent sur votre ordinateur.
- Cliquez sur le bouton **Référence Excel** pour sélectionner une cellule qui sera évaluée au nom et au chemin d'un classeur.
- Entrez manuellement un nom et chemin de classeur.
- Entrez manuellement une expression Excel qui sera évaluée à un chemin et un nom de classeur.

Le chemin d'un classeur dans l'arborescence des applications TM1 utilise le format suivant :

<NomDossier>\<NomDossier>\<NomClasseur>

Par exemple :

Planning Sample\Bottom Up Input\Budget Input

Le chemin du fichier réseau utilise le format :

\\<NomOrdinateur>\<NomDossier>\<NomClasseur>

Par exemple :

\\boston\reports\2007\_summary.xls

Pour des détails et des exemples, reportez-vous à la documentation IBM Cognos *TM1 for Developers*.

### Feuille

Contient le nom de la feuille de calcul à laquelle vous souhaitez accéder. Vous pouvez entrer cette valeur de l'une des manières suivantes :

- Cliquez sur le bouton **Parcourir** pour sélectionner un classeur, puis une feuille de calcul dans la liste Feuille.
- Entrez manuellement un nom de feuille de calcul.

- Entrez manuellement une référence Excel qui sera évaluée à un chemin et un nom de feuille de calcul.
- Cliquez sur le bouton **Référence Excel** pour sélectionner une cellule qui sera évaluée au nom d'une feuille de calcul.

Pour des détails et des exemples, reportez-vous à la documentation IBM Cognos *TM1 for Developers*.

### Apparier les éléments en titre

Cette option apparie automatiquement et définit les dimensions en titre entre la feuille de calcul source et la feuille de calcul cible quand un utilisateur clique sur le bouton **Action** pour naviguer jusqu'à la feuille de calcul cible.

Pour des détails et des exemples, reportez-vous à la documentation IBM Cognos *TM1 for Developers*.

### Remplacer le classeur en cours

Cette option détermine le mode d'ouverture de la feuille de calcul cible.

- Si cette option n'est pas sélectionnée (valeur par défaut), la feuille de calcul cible est ouverte dans une nouvelle fenêtre, dans Excel, ou dans un nouvel onglet, dans TM1 Web.
- Si cette option est sélectionnée, la feuille de calcul cible est ouverte dans la même fenêtre ou le même onglet et remplace la feuille de calcul source.



**ATTENTION** : Si vous activez cette option, n'oubliez pas d'enregistrer votre classeur *avant* de tester le nouveau bouton. Vous pourriez perdre vos modifications si vous cliquez sur le nouveau bouton et fermez le classeur en cours.

### Options avancées

Cliquez sur ce bouton pour ouvrir la boîte de dialogue Options avancées dans laquelle vous pouvez mapper manuellement les Zones entre la feuille de calcul source et la feuille de calcul cible pour un bouton d'action qui vous permet de passer d'une feuille de calcul à une autre.

Pour plus d'informations, voir [«Boîte de dialogue Options avancées»](#), à la page 5.

## Onglet Apparence

Servez-vous de l'onglet Apparence pour configurer l'apparence du bouton d'action.

### Légende

Définit le texte de légende qui s'affiche sur le bouton d'action.

### Police

Cliquez sur ce bouton pour afficher la boîte de dialogue Police dans laquelle vous pouvez définir le style et la taille de la police pour le texte du bouton.

### Afficher l'image d'arrière-plan

Permet de sélectionner un fichier image (en format bmp, gif ou jpg) qui s'étendra pour coïncider avec le bouton d'action.

Sélectionnez cette option puis cliquez sur **Parcourir** pour identifier et sélectionner le fichier image que vous souhaitez utiliser.

### Afficher comme lien hypertexte

Quand elle est activée, cette option affiche le bouton d'action sous forme de lien hypertexte avec du texte souligné bleu au lieu d'un bouton standard.

Cette option n'est pas disponible quand vous sélectionnez l'option Afficher l'image d'arrière-plan.

### Aperçu

Cette zone affiche un aperçu du texte de la légende, du style et de la couleur de la police et de la couleur d'arrière-plan du bouton.

### Couleurs

Permet de définir les couleurs du texte et de l'arrière-plan du bouton d'action.

Cliquez sur l'échantillon de couleur Texte ou Arrière-plan pour afficher la boîte de dialogue Couleur dans laquelle vous pouvez sélectionner une couleur standard ou définir une couleur personnalisée.

Cette option n'est pas disponible quand vous sélectionnez l'option Afficher comme lien hypertexte.

## Boîte de dialogue Options avancées

Servez-vous de la boîte de dialogue Options avancées pour mapper manuellement les zones entre la feuille de calcul source et la feuille de calcul cible quand vous insérez un bouton d'action qui vous permet de passer d'une feuille de calcul à une autre. Cet outil vous aide à mapper les dimensions, les cellules et les valeurs de la feuille de calcul source sur la feuille de calcul cible.

**Remarque :** Le mappage avancé est appliqué après tout mappage automatique effectué par l'option Appariement des éléments en titre.


Zone	Description
Ajouter	Ajoute une nouvelle ligne à la grille Mappage avancé.
Supprimer	Supprime la ligne sélectionnée dans la grille Mappage avancé.
OK	Ferme la boîte de dialogue Options avancées et enregistre vos réglages.
Annuler	Ferme la boîte de dialogue Options avancées sans enregistrer vos réglages.

Pour des exemples d'utilisation de la boîte de dialogue Options avancées, reportez-vous à *TM1 for Developers* dans l'[IBM Knowledge Center](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0) ([https://www.ibm.com/support/knowledgecenter/SSD29G\\_2.0.0](https://www.ibm.com/support/knowledgecenter/SSD29G_2.0.0)).

### Grille de mappage avancé

La boîte de dialogue Options avancées comprend une grille dans laquelle vous définissez le mappage des zones entre la feuille de calcul source et la feuille de calcul cible. Vous pouvez utiliser la grille pour spécifier comment les éléments de la feuille de calcul source et de la feuille de calcul cible sont appariés quand la feuille cible est ouverte. Chaque ligne de la grille définit une configuration de mappage.

Zone	Description
Type de source	<p>Cette zone représente le <i>type</i> d'objet pour la valeur que vous souhaitez mapper.</p> <p>Sélectionnez le Type de source comme suit :</p> <ul style="list-style-type: none"><li>• SUBNM - Indique que vous mappez à partir d'une cellule contenant une dimension en titre dans la feuille de calcul source.</li><li>• DBRW sélectionnée - Indique que vous mappez à partir d'une cellule contenant une formule DBRW dans la feuille de calcul source.</li><li>• Valeur - Indique que vous allez saisir une chaîne ou une valeur numérique qui sera envoyée à la cible.</li></ul>

Zone	Description
Objet source	<p>La valeur de cette zone dépend de ce qui est sélectionné dans la zone Type de source.</p> <p>Entrez l'Objet source comme suit :</p> <ul style="list-style-type: none"> <li>• Si le Type de source est défini à SUBNM, vous devez spécifier le nom de la dimension en titre qui existe dans la feuille de calcul source.</li> <li>• Si Type de source est défini à DBRW sélectionnée, vous devez spécifier le nom d'une dimension en titre en ligne ou en colonne existant dans la feuille de calcul source.</li> <li>• Si Type de source est défini à Valeur, vous devez entrer une chaîne ou une valeur numérique qui sera envoyée à la feuille de calcul cible.</li> </ul> <p>Vous pouvez aussi extraire ces valeurs de la feuille de calcul source en utilisant le symbole = pour créer une référence Excel.</p>
Type de cible	<p>Cette zone est le <i>type</i> de cellule de la feuille de calcul cible dans laquelle la valeur provenant de la zone Objet source sera insérée.</p> <p>Sélectionnez le Type de cible comme suit :</p> <ul style="list-style-type: none"> <li>• SUBNM - Indique que la cible est une dimension en titre dans la feuille de calcul cible.</li> <li>• Plage nommée - Indique que la cible est une plage nommée dans la feuille de calcul cible.</li> <li>• Plage - Indique que l'emplacement cible est une cellule de la feuille de calcul cible.</li> </ul> <p> <b>ATTENTION</b> : Si vous définissez le Type de cible à Plage nommée ou Plage, toute donnée ou formule préexistante se trouvant dans la cellule cible sera remplacée si vous utilisez le bouton d'action pour naviguer. Si la cellule cible contient une fonction TM1 DBRW, cette fonction sera perdue et la cellule ne pourra plus se connecter au serveur ou y lire ou écrire des données.</p>

Zone	Description
Objet cible	<p>Cette zone est l'emplacement dans la feuille de calcul cible où la valeur provenant de la zone Objet source sera insérée.</p> <p>Entrez l'Objet cible comme suit, en fonction de ce que vous avez sélectionné comme Type de cible :</p> <ul style="list-style-type: none"> <li>• Si le Type de cible est défini à SUBNM, vous devez spécifier le nom de la dimension en titre de la feuille de calcul cible.</li> <li>• Si le Type de cible est défini à Plage nommée, vous devez spécifier le nom de la plage dans la feuille de calcul cible.</li> <li>• Si le Type de cible est défini à Plage, vous devez spécifier l'emplacement de la cellule dans la feuille de calcul cible.</li> </ul> <p>Vous pouvez aussi utiliser une référence Excel pour extraire la valeur de la zone Objet cible.</p> <p>Pour un exemple détaillé, reportez-vous à la documentation <i>IBM CognosTM1 for Developers</i>.</p>
Sous-ensemble	Entrez une valeur pour la zone Sous-ensemble quand le Type de cible est défini à SUBNM.
Alias	Entrez une valeur pour la zone Alias quand le Type de cible est défini à SUBNM.

## Editeur d'attributs

Utilisez l'Editeur d'attributs pour créer et modifier les attributs des cubes, dimensions, éléments et répliquations.

Notez que tous les éléments comprennent un attribut Format définissant leur affichage dans le Cube Viewer. La valeur par défaut de l'attribut Format est Sans style.

### Menu Fichier

Élément de menu	Description
Fermer	Ferme l'Editeur d'attribut.

### Menu Edition

Élément de menu	Description
Annuler la cellule	Annule la dernière action sur la cellule. Cette option ne s'applique qu'aux cellules individuelles. Il est impossible d'annuler des actions appliquées à une plage de cellules.

Élément de menu	Description
Couper	Coupe le contenu des cellules sélectionnées et le place dans le Presse-papiers.
Copier	Copie le contenu des cellules sélectionnées dans le Presse-papiers.
Coller	Colle le contenu du Presse-papiers dans les cellules sélectionnées.
Ajouter un nouvel attribut	Ouvre la boîte de dialogue Nouvel attribut qui permet de créer un nouvel attribut pour les éléments de la dimension.
Supprimer l'attribut sélectionné	Supprime un attribut sélectionné. Vous devez supprimer les attributs individuellement. Il est impossible de supprimer plusieurs attributs simultanément.
Effacer	Efface le contenu des cellules sélectionnées.
Editer le format d'élément	Ouvre la boîte de dialogue Format des nombres dans laquelle vous pouvez affecter des valeurs d'attribut de Format.

## Options de mise en forme

L'option de mise en forme est disponible quand vous sélectionnez des cellules à l'intersection de la colonne Format et de lignes d'éléments. Cliquez sur le bouton **Format** pour afficher la boîte de dialogue Format des nombres.

Sélectionnez une option dans la zone de liste **Catégorie** pour spécifier le format d'affichage des cellules sélectionnées.

Les formats de nombres suivants sont disponibles :

Catégorie de format	Description
Général	<p>Ce format affiche les nombres sans virgule pour séparer les décimales (situées à gauche). Les valeurs négatives sont précédées du signe moins (-).</p> <p>Utilisez l'option Précision pour spécifier le nombre de décimales après la virgule décimale. Notez que les valeurs dérivées de règles ne renvoient des entiers que lorsque le format Général a été choisi.</p>
Fixe	<p>Ce format affiche les nombres sans virgule pour séparer les décimales (situées à gauche). Les valeurs négatives sont précédées du signe moins (-) ; les utilisateurs peuvent utiliser des parenthèses pour les nombres négatifs s'ils le préfèrent.</p> <p>Utilisez l'option Précision pour spécifier le nombre de décimales après la virgule décimale.</p>



Catégorie de format	Description
Devise	Ce format affiche les nombre avec le symbole de monnaie défini dans les Propriétés de Paramètres régionaux de Windows et utilise un espace pour séparer les groupes de 3 chiffres à gauche de la virgule décimale. Les valeurs négatives sont précédées du signe moins (-).  Utilisez l'option Précision pour spécifier le nombre de décimales après la virgule décimale.
Date	Affiche une liste de formats de date prédéfinis.
Période	Affiche une liste de formats d'heure prédéfinis.
Pourcentage	Ce format multiplie les nombres par 100 et affiche un signe de pourcentage (%) à la fin. Les chiffres à gauche de la virgule décimale n'utilisent pas de séparateur et les valeurs négatives sont précédées du signe moins (-).  Utilisez l'option Précision pour spécifier le nombre de décimales après la virgule décimale.
Scientifique	Ce format affiche les nombres en notation scientifique. Les valeurs négatives sont précédées du signe moins (-).  Utilisez l'option Précision pour spécifier le nombre de décimales après la virgule décimale.
Personnalisé	Vous pouvez définir une expression de format personnalisé selon les besoins.
Précision	Cette option détermine le nombre de décimales à afficher pour un format sélectionné. Si une valeur a plus de chiffres décimaux que la précision spécifiée, elle est arrondie pour l'affichage seulement. La valeur entière est stockée dans la base de données TM1.


## Fenêtre Journal d'audit

La fenêtre Journal d'audit permet d'interroger et d'afficher les enregistrements contenus dans le journal d'audit de TM1.

La fenêtre Journal d'audit contient deux panneaux principaux : le panneau de requête et le panneau de résultats. Servez-vous de ces panneaux pour effectuer des recherches dans le journal d'audit et afficher les enregistrements retrouvés par votre recherche.

### Panneau de requêtes

Le panneau de requêtes permet de générer des requêtes qui effectuent des recherches dans le journal d'audit de TM1.



La barre d'outil du panneau de requêtes comporte une icône **Exécuter la requête**  qui interroge le journal d'audit une fois que vous avez défini les options de requête.

Ces options sont regroupées de la façon suivante :

- Date et heure
- Propriétaire de l'événement
- Type d'événement

### Options de date et heure

Les options Date et heure comprennent la définition de la période sur laquelle la requête doit porter.



Option	Description
Période	<p>Contient une liste de périodes prédéfinies pour la requête.</p> <p>Sélectionnez une période prédéfinie ou sélectionnez <b>Période personnalisée</b> pour activer les options de période Début et Fin.</p>
Heure de début	<p>La date et l'heure de début de la requête.</p> <p>Cette option est activée uniquement quand vous sélectionnez <b>Période personnalisée</b> comme option de Période.</p> <p>TM1 interroge tous les enregistrements consignés au journal d'audit à partir de ces date et heure.</p> <p>Cliquez sur  pour ouvrir un calendrier permettant de spécifier une date et une heure.</p>
Heure de fin	<p>Date et heure de fin de la requête.</p> <p>Cette option est activée uniquement quand vous sélectionnez <b>Période personnalisée</b> comme option de Période.</p> <p>TM1 interroge tous les enregistrements d'audit jusqu'aux date et heure de fin que vous spécifiez.</p> <p>Cliquez sur l'icône Calendrier  pour ouvrir un calendrier permettant de sélectionner une date et une heure.</p> <p>Par défaut, la date et l'heure de fin sont la date et l'heure en cours.</p>

### Options Propriétaire de l'événement

Les options Propriétaire de l'événement répondent à la question "Qui a causé cet événement". Le propriétaire de l'événement peut être un véritable utilisateur de TM1 ou une tâche planifiée.

Les options Propriétaire de l'événement comprennent les paramètres suivants :


Option	Description
Tout	Définit la requête de recherche de tous les événements d'audit provoqués par un utilisateur de TM1 ou une tâche planifiée.

Option	Description
Client	<p>Définit la requête de tous les événements d'audit provoqués uniquement par des utilisateurs de TM1.</p> <p>Pour rechercher les événements provoqués par un utilisateur spécifique de TM1, cliquez sur le bouton <b>Sélectionner un client</b> . Vous pouvez sélectionner un client unique ou plusieurs.</p> <p>Le réglage par défaut est tous les clients.</p>
Tâche planifiée	<p>Définit la requête de tous les événements d'audit provoqués uniquement par des tâches planifiées.</p> <p>Pour rechercher les événements engendrés par une tâche planifiée spécifique, cliquez sur le bouton <b>Sélectionner une tâche planifiée</b> .</p> <p>Vous pouvez sélectionner une seule tâche planifiée ou plusieurs.</p> <p>Le réglage par défaut est toutes les tâches planifiées.</p>

### Options Type d'événement

Les options Type d'événement vous permettent de sélectionner les types d'objet ou d'événement que vous voulez rechercher. Par exemple, vous pouvez utiliser ces options de recherche pour trouver des tentatives de connexion échouées ou des événements pendant lesquels une dimension a été supprimée.

Option	Description
Tout	<p>Définit la requête de recherche des deux types d'événement d'audit : dans tout le système ou relatifs à des objets.</p>
Dans tout le système	<p>Définit la requête pour rechercher uniquement les événements d'audit dans tout le système.</p> <p>Pour rechercher un événement spécifique dans tout le système, sélectionnez cet événement dans la liste.</p> <p>Le paramètre par défaut est la recherche de <i>tous</i> les événements dans tout le système.</p>



Option	Description
Objet	<p>Définit la requête pour rechercher les événements d'audit uniquement de type objet.</p> <p>Pour rechercher un événement spécifique, utilisez les options comme suit :</p> <ul style="list-style-type: none"> <li>• <b>Type d'objet</b> - Limite la requête à un type spécifique d'objet TM1. Par exemple, seulement les événements relatifs aux dimensions.</li> <li>• <b>Nom d'objet</b> - Vous permet de sélectionner un nom d'objet spécifique.</li> </ul> <p>Cliquez sur  pour afficher une boîte de dialogue permettant de sélectionner des objets par nom.</p> <p><b>Remarque :</b> Lorsque vous réglez l'option Type d'objet sur <b>Elément</b>, le bouton Sélection de nom d'objet est désactivé car la liste d'éléments risque d'être trop longue à afficher. Pour rechercher les événements relatifs à un élément spécifique, vous devez entrer un nom d'élément manuellement en utilisant le format suivant : DimensionName:ElementName. Par exemple : region:italy</p> <ul style="list-style-type: none"> <li>• <b>Type d'événement</b> - Limite la requête à un type spécifique d'événement d'objet. Le paramètre par défaut est la recherche de <i>tous</i> les événements dans tout le système.</li> </ul>


## Panneau Résultats

Servez-vous du panneau Résultats pour afficher et naviguer dans les enregistrements retrouvés par votre recherche.

### Barre d'outils de la sous-fenêtre Résultats


La barre d'outils Résultats contient les boutons suivants :

Action	Bouton	Description
Copier		Copie la valeur de la cellule sélectionnée dans le presse-papiers de Windows.
Rechercher		Ouvre la boîte de dialogue Rechercher dans laquelle vous pouvez faire des recherches de texte dans les enregistrements d'événements.

Action	Bouton	Description
Exporter		Ouvre la boîte de dialogue Enregistrer sous qui vous permet d'enregistrer les événements dans un fichier dans l'un des formats suivants : <ul style="list-style-type: none"> <li>• XML</li> <li>• Séparé par des virgules</li> <li>• Séparé par des tabulations</li> </ul>

### Grille de résultats

La sous-fenêtre Résultats contient une grille qui affiche les enregistrements du journal d'audit retrouvés par la requête. Les enregistrements retrouvés sont organisés dans les colonnes suivantes :

En colonnes	Description
Date	Date et heure de l'événement
Utilisateur	Client TM1 (utilisateur) ou tâche planifiée qui a provoqué l'événement.
Type d'événement/Description	Brève description de l'événement.
Type d'objet	Type d'objet TM1 associé à l'événement.
Nom de l'objet	Nom de l'objet TM1 associé à l'événement.
Détails	Affiche une icône indiquant qu'il existe des informations détaillées portant sur cet événement spécifique. S'il existe des détails pour un événement, vous pouvez les consulter en cliquant sur l'icône <b>Détails</b>  de cet enregistrement.


Il est possible de trier les enregistrements contenus dans chaque colonne de la grille en ordre croissant ou décroissant en cliquant sur le titre de la colonne.



## Fenêtre Détail du journal d'audit

La fenêtre Détails du journal d'audit affiche les sous-événements pour un événement de journal d'audit qui était affiché dans les résultats de requête de la fenêtre Journal d'audit principale

### Barre d'outils Détails

La barre d'outils Détails contient les boutons suivants.

Bouton	Description
Copier 	Copie la valeur de la cellule sélectionnée dans le presse-papiers de Windows.

Bouton	Description
Rechercher 	Ouvre la boîte de dialogue Rechercher dans laquelle vous pouvez faire des recherches de texte dans les enregistrements d'événements.
Exporter 	Ouvre la boîte de dialogue Enregistrer sous qui vous permet d'enregistrer les événements dans un fichier dans l'un des formats suivants : <ul style="list-style-type: none"> <li>• XML</li> <li>• séparé par des virgules</li> <li>• séparé par des tabulations</li> </ul>

## Grille Détails

La grille Détails affiche les enregistrements détaillés des sous-événements appartenant à l'événement parent.

Les enregistrements détaillés sont organisés dans les colonnes suivantes :

En colonnes	Description
Date	Date et heure de l'événement
Utilisateur	Client TM1 (utilisateur) ou tâche planifiée qui a provoqué l'événement.
Type d'événement/Description	Brève description de l'événement.
Type d'objet	Type d'objet TM1 associé à l'événement.
Nom de l'objet	Nom de l'objet TM1 associé à l'événement.

Il est possible de trier les enregistrements contenus dans chaque colonne de la grille en ordre croissant ou décroissant en cliquant sur le titre de la colonne.





## Assistant Définition de tâche

Servez-vous de l'Assistant Définition de tâche pour planifier une réplication ou un processus de façon à les synchroniser ou les exécuter à intervalle régulier.

L'Assistant est composé de deux écrans :

- **Ecran 1** - Sélectionnez les processus et les réplifications à inclure dans la tâche.
- **Ecran 2** - Spécifiez l'heure de début de l'exécution initiale de la tâche et l'intervalle d'exécution ultérieur de la tâche.

### Ecran 1 (Etape 1)

Zone	Description
Liste des disponibilités	Affiche une liste de toutes les réplifications et tous les processus disponibles pour la planification des tâches.
Liste des sélections	Affiche une liste des réplifications ou processus sélectionnés pour inclusion dans la tâche en cours.
Ajouter 	Cliquez sur ce bouton pour déplacer les processus ou réplifications sélectionnés de la liste Disponible à la liste Sélectionné.
Ajouter tout 	Cliquez sur ce bouton pour déplacer tous les processus ou réplifications de la liste Disponible à la liste Sélectionné.
Supprimer 	Cliquez sur ce bouton pour déplacer les processus ou réplifications sélectionnés de la liste Sélectionné à la liste Disponible.
Supprimer tout 	Cliquez sur ce bouton pour déplacer tous les processus ou réplifications de la liste Sélectionné à la liste Disponible.
Spécifier les valeurs des paramètres	Cliquez pour ouvrir la boîte de dialogue Valeurs de paramètre qui permet de spécifier des valeurs pour tous les paramètres associés au processus sélectionné.

## Ecran 2 (Etape 2)

Zone	Description
Date et heure de début de la tâche	Sélectionnez une date du calendrier pour spécifier une heure de début dans la zone Heure.
Fréquence d'exécution de la tâche	Remplissez les zones appropriées pour définir l'intervalle dans lequel la tâche doit être exécutée.
Le planning de tâches est actif	Remplissez cette case pour activer l'exécution de la tâche à l'heure de début et à l'intervalle spécifiés. Désactivez cette case pour activer la tâche ultérieurement.

## Fenêtre Clients/Groupes

La fenêtre Clients/Groupes permet de créer et modifier les clients et les groupes d'utilisateurs sur un serveur.

## Grille Clients/Groupes

La grille Clients/Groupes affiche le nom des clients comme en-têtes de ligne et celui des groupes d'utilisateurs comme en-têtes de colonne. Un X à l'intersection d'un nom de client et d'un groupe d'utilisateurs indique le groupe auquel appartient l'utilisateur. Les utilisateurs peuvent appartenir à plusieurs groupes.

Cette grille comprend aussi plusieurs colonnes qui affichent les propriétés des clients sur le serveur.

- La cellule se trouvant à l'intersection d'un nom de client et de la colonne Mot de passe contient le mot de passe de ce client.
- La cellule se trouvant à l'intersection d'un nom de client et de la colonne Jours de validité indique le nombre de jours pendant lequel le mot de passe est valide pour le client. Une fois ce nombre de jours écoulé, le client ne peut plus se connecter au serveur avec ce mot de passe. Un client dont le mot de passe va expirer sous peu commence à recevoir des avis cinq jours avant la date d'expiration.
- La cellule à l'intersection du nom du client et de la colonne Statut indique si le client est actif sur le serveur.
- La cellule à l'intersection du nom du client et de la colonne Nombre maximum de connexions indique le nombre maximum de connexion qui peuvent être établies avec le serveur à l'aide du nom de client et du mot de passe en question.

## Menu Sécurité

Élément de menu	Description
Fermer	Ferme la boîte de dialogue Clients/Groupes.

## Menu Clients

Élément de menu	Description
Ajouter un nouveau client	Ouvre la boîte de dialogue Création d'un nouveau client qui permet de créer un client sur le serveur.
Supprimer un client	Supprime le client actuellement sélectionné du serveur.
Déconnecter un client	Déconnecte le client actuellement sélectionné du serveur.
Définir un mot de passe	Définit le mot de passe du client actuellement sélectionné.
Effacer le mot de passe	Efface le mot de passe pour le client actuellement sélectionné.

## Menu Groupes



<b>Élément de menu</b>	<b>Description</b>
Ajouter un nouveau groupe	Ouvre la boîte de dialogue Création d'un nouveau groupe qui permet de créer un groupe d'utilisateurs sur le serveur.
Supprimer un groupe	Supprime le groupe actuellement sélectionné du serveur.

## Grille Clients/Groupes

Il est possible d'entrer des données pour les clients directement dans la grille Clients/Groupes.

Cette grille comprend plusieurs colonnes qui sont décrites dans le tableau ci-dessous.

<b>En colonnes</b>	<b>Description</b>
Nom d'utilisateur	Affiche les noms d'utilisateur de tous les clients sur le serveur.
Mot de passe	Identifie si un mot de passe a été défini pour un client donné.  Cliquez sur la colonne à l'intersection de la colonne Mot de passe et d'une ligne de client, puis tapez un mot de passe pour affecter un mot de passe à ce client.  Lorsque vous saisissez un mot de passe, TM1 vous demande de le retaper pour le confirmer.
Jours de validité	Indique le nombre de jours pendant lequel un mot de passe de client est valide.  Pour affecter un nombre de jours de validité pour un client, cliquez sur la cellule à l'intersection de la colonne Jours de validité et de la ligne du client, puis tapez une valeur de jours de validité.
Nombre maximum de connexions	Définit le nombre maximum de connexions que le serveur peut avoir avec un client donné.  Pour affecter un nombre de jours de validité pour un client, cliquez sur la cellule à l'intersection de la colonne Nombre maximum de connexions et de la ligne du client, puis tapez un nombre maximum de connexions.
Statut	Indique le statut de connexion en cours d'un client donné.

En colonnes	Description
Groupes d'utilisateurs	<p>Il existe une colonne pour chaque groupe d'utilisateurs sur le serveur.</p> <p>Pour affecter un client à un groupe d'utilisateurs, remplissez la case à cocher à l'intersection de la colonne du groupe d'utilisateurs et du nom du client.</p> <p>Les clients peuvent appartenir à plusieurs groupes d'utilisateurs.</p>

## Boîte de dialogue Centre de messagerie clients

La boîte de dialogue Centre de messagerie clients permet de gérer les connexions des clients à un serveur. Elle permet également d'arrêter un serveur à distance. Vous devez faire partie du groupe ADMIN d'un serveur pour pouvoir accéder à cette boîte de dialogue.

Sélectionnez un serveur dans la sous-fenêtre gauche de l'Explorateur de serveur, puis sélectionnez **Serveur, Gestionnaire de serveurs pour ouvrir la boîte de dialogue Centre de messagerie clients.**

Zone	Description
Arrêter le serveur	Sélectionnez cette option pour arrêter le serveur, puis spécifiez un intervalle en minutes.
Déconnecter des clients	<p>Sélectionnez cette option pour déconnecter des clients du serveur, puis spécifiez un intervalle en minutes.</p> <p>Vous devez cliquer sur Sélectionner des clients pour créer ou sélectionner un sous-ensemble de clients à déconnecter.</p>
Diffuser le message aux clients sélectionnés	<p>Sélectionnez cette option pour diffuser un message textuel aux clients connectés au serveur.</p> <p>Saisissez le message dans la zone de texte puis cliquez sur Sélectionner des clients pour créer ou sélectionner un sous-ensemble de clients devant recevoir le message.</p>

## Boîte de dialogue Créer une dimension.

Saisissez le nom de la dimension à créer dans la zone en haut de la boîte de dialogue, puis cliquez sur **OK**.

Pour créer une dimension sur le serveur local, entrez uniquement le nom de la dimension.

Pour créer une dimension sur un serveur distant, faites précéder le nom de la dimension du nom du serveur et d'un deux-points. Par exemple, saisissez **Ventes:Produit** pour créer la dimension Produit sur le serveur Ventes.

## Boîte de dialogue Créer un objet de réplication de serveur

Utilisez la boîte de dialogue Créer un objet de réplication de serveur pour créer une nouvelle connexion de réplication ou pour en modifier une existante.

Zone	Description
Vers le serveur	Sélectionnez un serveur source dans la liste. Cette liste contient le nom de tous les serveurs actuellement disponibles sur votre réseau.
Comme utilisateur	Entrez votre nom d'utilisateur sur le serveur source sélectionné.
Avec le mot de passe	Saisissez votre mot de passe pour le serveur source sélectionné.
Avec espace-noms	Si l'objet utilise la sécurité CAM Passport, entrez l'ID d'espace-noms IBM Cognos. N'utilisez pas le nom descriptif ici.
Utiliser la connexion intégrée	Cochez cette case pour utiliser l'authentification par connexion intégrée au lieu de la sécurité TM1 standard.

## Boîte de dialogue Créer un cube

Utilisez les options suivantes de la boîte de dialogue Créer un cube pour créer un nouveau cube à partir de dimensions définies précédemment.

Zone	Description
Nom du cube	Tapez le nom du cube que vous créez dans cette zone.
Dimensions disponibles	Une liste des dimensions disponibles sur le serveur où vous créez le cube.
Dimensions du nouveau cube	La liste de dimensions du cube que vous créez.
Ajouter	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées de la liste Dimensions disponibles à la liste Dimensions du nouveau cube.
Supprimer	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées de la liste Dimensions du nouveau cube à la liste Dimensions disponibles.
Déplacer vers le haut	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées vers le haut dans la liste Dimensions du nouveau cube. Chaque clic du bouton déplace les dimensions sélectionnées d'une position vers le haut.

Zone	Description
Déplacer vers le bas	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées vers le bas dans la liste Dimensions du nouveau cube. Chaque clic du bouton déplace les dimensions sélectionnées d'une position vers le bas
Annuler	Cliquez pour annuler la création de cube et quitter la boîte de dialogue Créer un cube.
Réinitialiser	Cliquez pour réinitialiser la liste Dimensions disponibles et effacer le contenu de la liste Dimensions du nouveau cube.
Actualiser	Cliquez pour actualiser la liste Dimensions disponibles. Cette option interroge le serveur pour y trouver toute nouvelle dimension et ajoute ces dimensions dans la liste Dimensions disponibles.
Propriétés	Cliquez sur ce bouton pour affecter des propriétés de cube.  Vous pouvez affecter des propriétés qui définissent une dimension de mesures, une dimension de temps et un statut de chargement sur demande pour le cube.
OK	Cliquez pour accepter la configuration de la boîte de dialogue et créer le cube.

## Boîte de dialogue Optimiseur de cube

TM1 inclut une fonction qui permet d'optimiser l'ordre des dimensions dans un cube pour réduire la consommation de la mémoire et améliorer les performances.

Si vous ne connaissez pas bien vos données, vous risquez d'indiquer lors de la création du cube un ordre des dimensions qui entraînera des performances décevantes. Il est possible que la répartition des données dans un cube évolue avec le temps et rende inefficace l'ordre défini lors de la création du cube.

Lorsque vous optimisez l'ordre des dimensions d'un cube, TM1 ne change *pas* l'ordre réel des dimensions dans la structure du cube. TM1 *change* l'organisation interne des dimensions sur le serveur, mais puisque la structure de cube n'est pas modifiée, les règles, fonctions ou applications référençant le cube restent valides.

A mesure que vous changez l'ordre des dimensions, vous pouvez visualiser instantanément un rapport faisant état de l'impact que ces changements ont sur l'utilisation de mémoire faite par le cube.

Pour les raisons qui suivent, vous ne devez optimiser l'ordre des dimensions d'un cube que dans un environnement de développement pour essayer de déterminer la configuration optimale du cube.

- Le serveur utilise des ressources de mémoire importantes pour reconfigurer l'organisation des dimensions dans un cube. Au cours de ce processus, la mémoire vive temporaire du serveur est doublée pour le cube en cours de réorganisation. Par exemple, la reconfiguration d'un cube de 50 Mo requiert 100 Mo de mémoire vive.
- Pendant la réorganisation, le serveur est en lecture seule et toutes les requêtes des utilisateurs sont verrouillées.

**Remarque :** Vous devez appartenir au groupe ADMIN pour optimiser l'ordre des dimensions dans un cube. L'option d'optimisation n'est disponible que pour les cubes sur des serveurs distants ; vous ne pouvez pas optimiser l'ordre des dimensions de cubes sur un serveur local. Notez aussi que, lorsque vous optimisez l'ordre des dimensions dans un cube, vous ne devez pas déplacer les chaînes *depuis* la dernière position et ne devez pas non plus les déplacer *vers* la dernière position.

### Procédure

1. Dans la sous-fenêtre de l'arborescence de l'Explorateur de serveur, sélectionnez le cube à optimiser.
2. Sélectionnez **Cube, Réorganiser les dimensions**.  
La boîte de dialogue **Optimiseur de cube** apparaît.
3. Dans la zone de liste **Nouvel ordre des dimensions**, sélectionnez une dimension.
4. Cliquez sur les flèches vers le haut ou vers le bas pour modifier l'ordre des dimensions dans le cube.
5. Cliquez sur l'option **Test**.

Notez la valeur indiquée à côté de Changement en pourcentage. Si cette valeur est négative, le nouvel ordre des dimensions utilise moins de mémoire et est par conséquent plus efficace.

6. Répétez les étapes 3 à 5 jusqu'à ce que vous ayez obtenu l'ordre optimal pour les dimensions.
7. Cliquez sur **OK**.

## Boîte de dialogue Propriétés de cube

Servez-vous de la boîte de dialogue Propriétés de cube pour définir les propriétés de cubes précis.

Zone	Description
Dimension de mesures	Sélectionnez une dimension de mesures dans la liste.
Dimension de temps	Sélectionnez une dimension de temps dans la liste.
Charger sur demande	Remplissez cette case pour charger le cube dans la mémoire du serveur seulement quand les données de cube sont requises par un client. Videz cette case pour charger le cube automatiquement au démarrage du serveur.

## Cube Viewer

### Dimensions en titre

Les dimensions en titre apparaissent directement sous la barre d'outils en haut de la fenêtre Cube Viewer. Chaque dimension apparaît dans une zone de liste.

### Dimensions en ligne

Les dimensions en ligne apparaissent en haut de l'axe des lignes du Cube Viewer. Les éléments de la dimension en cours apparaissent comme en-têtes de ligne dans le Cube Viewer.

## Dimensions en colonne

Les dimensions en colonne apparaissent à gauche de l'axe des colonnes du Cube Viewer. Les éléments de la dimension en cours apparaissent comme en-têtes de colonne dans le Cube Viewer.

## Menu Fichier

Les options suivantes sont disponibles dans le menu Fichier du Cube Viewer.

Option	Description
Ouvrir	Ouvre la boîte de dialogue Ouvrir une vue de TM1, qui permet d'ouvrir d'autres vues associées au cube actuel.
Recharger	Recharge la définition de vue en cours.
Calculer	Calcule la vue en cours.
Enregistrer	Enregistre la définition de vue en cours.
Enregistrer sous	Enregistre la définition de vue en cours sous un nouveau nom.
Supprimer les vues	Ouvre la boîte de dialogue Supprimer les vues nommées qui permet de supprimer des vues enregistrées.
Tranche	Exporte la vue en cours dans une feuille de calcul Excel. La feuille de calcul Excel est remplie avec des formules qui récupèrent des valeurs du serveur dont la vue provient et y écrivent des données.
Formulaire actif	Lance l'option Insérer un formulaire actif qui vous permet d'ajouter une connexion Formulaire actif aux données de la cellule en cours de la feuille de calcul.
Instantané	Exporte la vue en cours dans une feuille de calcul Excel sous forme de valeurs simples. La feuille de calcul ne conserve pas de connexion au serveur d'où la vue provient.
Fermer	Ferme la fenêtre du Cube Viewer.

## Menu Edition

Les options suivantes sont disponibles dans le menu Edition du Cube Viewer.

Option	Description
TransAction	Annule la dernière action sur la cellule. Les options Enregistrer ou Fermer mettent fin à l'ensemble d'actions qu'il est possible d'annuler ou de rétablir.  L'option Rétablir permet de restaurer la dernière action sur la cellule.

Option	Description
Couper	Coupe le contenu des cellules sélectionnées et le place dans le Presse-papiers.
Copier	Copie le contenu des cellules sélectionnées, dans leur format en cours, dans le Presse-papiers.
Copier la valeur non formatée	Copie le contenu des cellules non formatées dans le Presse-papiers.
Coller	Colle le contenu du Presse-papiers dans les cellules sélectionnées.
Supprimer	Supprime les valeurs des cellules sélectionnées.
Editer les attributs de cube	Ouvre la fenêtre de l'Editeur d'attributs qui permet d'affecter et modifier les attributs de tous les cubes du serveur en cours.

## Menu Affichage

Les options suivantes sont disponibles dans le menu Affichage du Cube Viewer.

Option	Description
Barre d'outils	Masque ou affiche la barre d'outils en haut du Cube Viewer. Une coche indique que la barre d'outils est affichée.
Barre de statut	Masque ou affiche la barre de statut au bas du Cube Viewer. Une coche indique que la barre de statut est affichée.
De droite à gauche	Ce bouton-bascule change la position des dimensions en colonne dans le Cube Viewer.  Une flèche pointant vers la droite indique que les colonnes sont disposées de droite à gauche. Une flèche pointant vers la gauche indique que les colonnes sont disposées de gauche à droite.

## Menu Options

Les options suivantes sont disponibles dans le menu Options du Cube Viewer.

Option	Description
Supprimer les valeurs nulles	Cette option supprime ou active l'affichage de toutes les lignes et colonnes contenant uniquement des valeurs zéro dans une vue de cube. Une coche indique que les lignes et les colonnes contenant uniquement des valeurs nulles sont supprimées dans la vue en cours.

Option	Description
Supprimer les zéros dans les lignes	Cette option supprime ou active l'affichage de toutes les lignes contenant uniquement des valeurs zéro dans la vue de cube. Une coche indique que les lignes contenant uniquement des valeurs nulles sont supprimées dans la vue en cours.
Supprimer les zéros dans les colonnes	Cette option supprime ou active l'affichage de toutes les colonnes contenant uniquement des valeurs zéro dans la vue de cube. Une coche indique que les colonnes contenant uniquement des valeurs nulles sont supprimées dans la vue en cours.
Recalcul automatique	Cette option active ou désactive le recalcul automatique lors de la redéfinition de vue. Une coche indique que la vue est recalculée automatiquement chaque fois que la définition de vue change.
Format	Ouvre la boîte de dialogue Format de nombres qui permet de définir le format des nombres pour les valeurs de la vue en cours. Notez que le format que vous sélectionnez ne s'applique qu'aux valeurs pour lesquelles il n'y a pas d'attribut de format sélectionné.
Largeur de colonne	Ouvre la boîte de dialogue Largeur de colonne qui permet de définir une largeur minimum pour les colonnes du Cube Viewer.
Faire une tranche dans un nouveau classeur	<p>Cette option détermine comment les tranches sont créées.</p> <p>Une coche indique que les tranches sont insérées dans un nouveau classeur quand vous sélectionnez Fichier, Tranche.</p> <p>Si cette option n'est pas activée, les tranches sont insérées dans une nouvelle feuille du classeur en cours.</p>

## Boîte de dialogue Supprimer les sous-ensembles nommés

La boîte de dialogue affiche les sous-ensembles associés à la dimension en cours. Pour supprimer un sous-ensemble, sélectionnez-le et cliquez sur OK.

Pour sélectionner plusieurs sous-ensembles adjacents, cliquez et faites glisser la souris sur ces sous-ensembles. Pour sélectionner plusieurs sous-ensembles non adjacents, appuyez sur Ctrl et cliquez sur chaque sous-ensemble.



## Boîte de dialogue Supprimer les vues nommées

---

La boîte de dialogue affiche les vues associées au cube en cours. Pour supprimer une vue, sélectionnez-la et cliquez sur OK.

Pour sélectionner plusieurs vues adjacentes, cliquez et faites glisser la souris sur les différentes vues. Pour sélectionner plusieurs vues non adjacentes, appuyez sur Ctrl et cliquez sur chaque vue.

## Editeur de dimension

---

### Sous-fenêtre Eléments

Affiche les éléments de la dimension actuellement affichée.

### Sous-fenêtre Propriétés

Quand vous sélectionnez un élément consolidé dans la sous-fenêtre Eléments, la sous-fenêtre Propriétés affiche les propriétés des enfants immédiats de l'élément consolidé.

Quand vous sélectionnez un élément feuille, la sous-fenêtre Propriétés en affiche les propriétés.

**Remarque :** Lorsque vous affichez un ensemble de dimensions de taille exceptionnelle dans l'Editeur de dimension et que la sous-fenêtre de Propriétés est active, il se peut que vous rencontriez des problèmes de performance. Cela peut se produire lorsque vous sélectionnez une consolidation dans la sous-fenêtre des éléments et que TM1 doit afficher la liste complète des propriétés et éléments apparentés dans la sous-fenêtre Propriétés.

Si vous travaillez avec des ensembles de dimensions de grande taille, vous voudrez peut-être désactiver la sous-fenêtre Propriétés. Pour désactiver la sous-fenêtre Propriétés, cliquez sur l'option **Fenêtre de propriétés** dans le menu Affichage pour supprimer la coche qui se trouve à côté de l'option.

## Menu Dimension

Élément de menu	Description
Enregistrer	Enregistre la structure de dimension en cours.
Enregistrer sous	Enregistre la structure de dimension en cours sous un nouveau nom.
Fermer	Ferme l'Editeur de dimension.

## Menu Edition

Élément de menu	Description
Couper	Coupe les éléments sélectionnés et les met dans le Presse-papiers.
Copier	Copie les éléments sélectionnés dans le Presse-papiers.

Élément de menu	Description
Coller	<p>Colle le contenu du Presse-papiers en dessous d'un nouvel élément.</p> <ul style="list-style-type: none"> <li>• Quand il n'y a pas d'éléments sélectionné dans l'Editeur de dimension, cette option insère un nouvel élément au-dessus du premier élément affiché dans la sous-fenêtre Eléments.</li> <li>• Quand un élément est sélectionné dans la sous-fenêtre Eléments, cette option affiche un sous-menu contenant les commandes Coller au-dessus, Coller comme enfant et Coller en dessous.</li> </ul>
Coller au-dessus	Colle le contenu du Presse-papiers au-dessus de l'élément sélectionné.
Coller en dessous	Colle le contenu du Presse-papiers en dessous de l'élément sélectionné.
Coller comme enfant	Colle le contenu du Presse-papiers au-dessus de l'élément sélectionné.
Insérer un enfant	Ouvre la boîte de dialogue Insertion d'élément de dimension qui permet d'insérer un ou plusieurs enfants d'un élément sélectionné.
Insérer un élément	Ouvre la boîte de dialogue Insertion d'élément de dimension qui permet d'insérer des éléments feuilles (simples) dans la dimension.
Sélectionner tout	Sélectionne tous les éléments de la sous-fenêtre Eléments.
Filtrer par niveau	<p>Ouvre la boîte de dialogue Filtrer par niveau qui permet de sélectionner des éléments par niveau de hiérarchie.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension. Quand vous sélectionnez cette option, la sous-fenêtre Eléments affiche seulement les éléments du niveau spécifié.</p>
Filtrer par attribut	<p>Ouvre la boîte de dialogue Filtrer par attribut qui permet de sélectionner des éléments par valeur d'attribut.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension. Quand vous sélectionnez cette option, la sous-fenêtre Eléments affiche seulement les éléments du niveau que vous avez spécifié.</p>

Élément de menu	Description
Filtrer par caractère générique	<p>Permet de sélectionner des éléments correspondant à une expression de recherche définie par l'utilisateur.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension. Quand vous sélectionnez cette option, la sous-fenêtre Eléments affiche seulement les éléments correspondant à l'expression de recherche que vous spécifiez.</p>
Sélectionner un alias	Ouvre la boîte de dialogue Alias de TM1, qui permet de sélectionner l'alias à afficher dans l'Editeur de dimension.
Trier par ordre croissant	<p>Trie tous les éléments de la sous-fenêtre Eléments par ordre alphabétique croissant.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.</p>
Trier par ordre décroissant	<p>Trie tous les éléments de la sous-fenêtre Eléments par ordre alphabétique décroissant.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.</p>
Trier par hiérarchie	<p>Trie tous les éléments dans la sous-fenêtre Eléments par ordre hiérarchique, pour afficher les relations parent/enfant entre les éléments.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.</p>
Trier par index croissant	<p>Trie tous les éléments de la sous-fenêtre Eléments par ordre croissant selon la valeur d'index d'élément.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.</p>
Trier par index décroissant	<p>Trie tous les éléments de la sous-fenêtre Eléments par ordre décroissant selon la valeur d'index d'élément.</p> <p>Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.</p>

<b>Élément de menu</b>	<b>Description</b>
Garder	Modifie la sous-fenêtre Éléments pour afficher seulement les éléments actuellement sélectionnés.  Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.
Masquer	Modifie la sous-fenêtre Éléments pour masquer les éléments actuellement sélectionnés.  Cette option n'affecte que l'affichage des éléments, elle n'affecte pas la structure de dimension.
Supprimer un élément	Supprime toutes les instances d'un élément sélectionné de la dimension.
Supprimer depuis la consolidation	Supprime l'instance d'un élément sélectionné de la consolidation en cours.
Editer des formats d'éléments	Ouvre la feuille de calcul Editer les formats d'éléments qui permet de définir les styles d'affichage des éléments. Ces styles d'affichage sont appliqués dans les tranches dynamiques et les feuilles Web TM1.
Développer un élément	Affiche tous les enfants d'un élément sélectionné.
Réduire un élément	Masque tous les enfants d'un élément sélectionné.
Propriétés	Ouvre la boîte de dialogue Propriétés d'élément de dimension qui permet d'affecter un type et une pondération à un élément sélectionné.

## Menu Affichage

<b>Élément de menu</b>	<b>Description</b>
Barres d'outils	Masque ou affiche les différentes barres d'outils en haut de la fenêtre de l'Editeur de dimension. Une coche indique qu'une barre d'outils est affichée.
Barre de statut	Masque ou affiche la barre de statut au bas de la fenêtre de l'Editeur de dimension. Une coche indique que la barre de statut est affichée.
Fenêtre Propriétés	Affiche ou masque la sous-fenêtre Propriétés. Une coche indique que la sous-fenêtre Propriétés est affichée.
Actualiser	Met à jour l'affichage de la sous-fenêtre Éléments.

## Boîte de dialogue Insertion d'élément de dimension

Utilisez cette boîte de dialogue pour ajouter des éléments simples, chaîne ou consolidés à une dimension. La boîte de dialogue contient les options suivantes.

Option	Description
Nom de la dimension	Le nom de la dimension à laquelle vous ajoutez des éléments. Cette option n'est pas modifiable.
Nom du parent	Le nom de l'élément parent auquel vous ajoutez des éléments. Cette option n'est pas modifiable.  Si un élément a été sélectionné dans l'Editeur de dimension lors de l'ouverture de la boîte de dialogue Insertion d'élément de dimension, il apparaît comme Nom du parent. Si aucun élément n'a été sélectionné, le Nom du parent est Racine.
Insérer un nom d'élément	Entrez un nom pour le nouvel élément dans cette zone.
Type d'élément	Opérez une sélection appropriée pour l'élément à insérer.
Pondération de l'élément	Si le type d'élément est Simple et que le Nom du parent est autre que Racine, entrez une pondération dans cette zone. La pondération est un facteur de multiplication appliqué à un élément lors de la consolidation.  Une pondération associée à un élément de consolidation ne change pas la valeur de l'élément ailleurs dans la dimension.
Ajouter	Cliquez sur Ajouter chaque fois que vous spécifiez un élément, type et pondération nouveau.
OK	Cliquez sur ce bouton quand vous avez fini d'ajouter des éléments pour les valider dans la dimension.

## Boîte de dialogue Ordre des éléments de dimension

Utilisez cette boîte de dialogue pour définir l'ordre des éléments d'une dimension.

L'ordre des éléments d'une dimension détermine la valeur d'index de chaque élément de la dimension. Le premier élément d'une dimension reçoit la valeur d'index 1, le second la valeur d'index 2, et ainsi de suite. L'ordre des éléments dans une dimension est important car un grand nombre de fonctions TM1 (feuille de calcul, règles et TurboIntegrator) référencent les valeurs d'index d'élément.

**Remarque :** Si vous changez l'ordre des éléments dans une dimension, toute fonction référençant des valeurs d'index d'élément renverra des valeurs nouvelles, voire inattendues.

Pour définir l'ordre des éléments, procédez comme suit.

## Procédure

1. Sélectionnez un type de tri.

Type	Description
Automatique	Affiche les options Trier automatiquement par : Nom, Niveau et Hiérarchie.
Manuel	Ordonne les éléments comme ils existent actuellement dans la structure de dimension et définit la propriété de tri de dimension à Manuel.

2. Si vous sélectionnez le type de tri Manuel, passez à l'étape 5.
3. Sélectionnez une option **Trier automatiquement par**.

Type	Description
Nom	Trie les éléments par ordre alphabétique.
Niveau	Trie les éléments par niveau de hiérarchie.
Hiérarchie	Trie les éléments selon la hiérarchie de dimension.

4. S'il y a lieu, sélectionnez une **Direction du tri**.
5. Cliquez sur **OK**.

Vous avez maintenant défini l'ordre des éléments de la dimension. Lorsque vous ouvrirez la dimension, vous verrez les éléments triés selon l'option Trier par que vous avez spécifiée à l'étape 3.

## Résultats

Vous avez maintenant défini l'ordre des éléments de la dimension. Lorsque vous ouvrirez la dimension, vous verrez les éléments triés selon l'option Trier par que vous avez spécifiée à l'étape 3.

## Boîte de dialogue Propriétés d'élément de dimension

Affiche le nom, le type et la pondération de l'élément en cours.

### Sous-fenêtre Propriétés

Options	Description
Type d'élément	Pour changer le type de l'élément en cours, sélectionnez un nouveau type dans la liste. Il existe trois types d'élément possibles : simple, consolidé et chaîne.
Pondération de l'élément	Pour changer la pondération de l'élément en cours, double-cliquez sur la zone Pondération de l'élément et saisissez une nouvelle pondération.

## Détailler

Le menu Détailler répertorie les options qui servent à créer un processus de détail et une affectation de détail. Les processus et affectations de détail permettent d'établir des liens entre les cellules d'un cube et les données détaillées qui leur sont associées.

Options	Description
Créer/Modifier/Supprimer des règles d'affectation de détail	Sélectionnez ces options pour créer, modifier ou supprimer des affectations de détail. L'option Créer ouvre l'Editeur de règles et vous permet de concevoir la règle.
Créer/Modifier le processus de détail	Un processus de détail est un processus TurboIntegrator qui définit les données détaillées, lesquelles sont ouvertes dans une nouvelle fenêtre. Ces options permettent de modifier une règle d'affectation de détail existante ou d'en créer une nouvelle. Les options Créer affichent les paramètres et les valeurs à utiliser, ainsi que les détails correspondant à la source de données. Si vous changez la source de données d'un processus de détail, TurboIntegrator ne met pas à jour la fonction avec la nouvelle source de données parce que la fonction se trouve hors de la zone <b>Generated Statements</b> . Vous devez modifier la source de données dans la fonction ReturnViewHandle pour le processus de détail.

## Boîte de dialogue Editer la formule

La boîte de dialogue Editer la formule vous guide lors de la création de fonctions DBR, DBRW et DBS. Vous pouvez également l'utiliser pour éditer une fonction TM1 dans une feuille de calcul.

Pour afficher la boîte de dialogue Editer la formule, cliquez sur une cellule de la feuille de calcul et sélectionnez **TM1, Editer la formule**. Si la cellule contient une fonction TM1, celle-ci s'affiche dans la zone de saisie de la boîte de dialogue.

Zone	Description
Réf. SGBD	Cliquez sur ce bouton pour insérer une fonction DBR dans la cellule en cours. TM1 vous guide dans une série de boîtes de dialogue qui vous aident à créer la fonction.
DBRW	Cliquez sur ce bouton pour insérer une fonction DBRW dans la cellule en cours. TM1 vous guide dans une série de boîtes de dialogue qui vous aident à créer la fonction.
Envoi SGBD	Cliquez sur ce bouton pour insérer une fonction DBS dans la cellule en cours. TM1 vous guide dans une série de boîtes de dialogue qui vous aident à créer la fonction.

Zone	Description
Réf. cellule	Cliquez sur ce bouton pour insérer une référence à une cellule dans une fonction. TM1 vous invite à sélectionner la cellule à laquelle vous souhaitez faire référence et à entrer un type de référence.
Noms	Cliquez sur ce bouton pour insérer un cube, une dimension ou un nom d'élément dans une fonction.

L'Editeur de formule vous aide à créer des fonctions référençant des cubes de 29 dimensions maximum.

## Boîte de dialogue Editer la référence au cube

Cette boîte de dialogue vous permet de définir les références d'élément utilisées dans les fonctions de feuille de calcul TM1 telles que DBRW et DBSW.

Cette boîte de dialogue contient des boutons et des zones correspondant à chaque dimension du cube référencé par la fonction de feuille de calcul TM1. Par exemple, l'image suivante illustre la boîte de dialogue Editer la référence au cube pour une fonction DBRW qui référence le cube SalesCube (Ventes) dans l'exemple de base de données TM1. Cette boîte de dialogue comprend des boutons pour toutes les dimensions du cube Ventes.

orsque vous insérez une fonction TM1 dans une feuille de calcul, TM1 tente de déterminer s'il existe dans la feuille de calcul des références d'élément pertinentes. Si c'est le cas, ces références sont insérées automatiquement dans les zones appropriées de la boîte de dialogue Editer la référence au cube. S'il est impossible de déterminer des références d'élément pertinentes, TM1 insère "Undef" dans ces zones.

Vous pouvez définir des références dans cette boîte de dialogue des manières suivantes :

- cliquer sur un bouton de dimension et sélectionner un élément. Dans ce cas, la référence est insérée sous forme de chaîne dans la zone appropriée.
- entrer une référence de cellule directement dans une zone. Vous pouvez utiliser des références relatives aux lignes ou colonnes, ou des références de cellule absolues.

Si le cube pour lequel vous créez une référence contient plus de 16 dimensions, cliquez sur **Précédent** pour retourner à la page contenant les 16 dimensions précédentes, ou sur **Suivant** pour avancer à la page contenant les 16 dimensions suivantes.

## Boîte de dialogue Filtrer les éléments par valeur d'attribut

Utilisez cette boîte de dialogue pour ne sélectionner que les éléments de sous-ensemble avec une valeur d'attribut spécifiée.

Sélectionnez un attribut dans la liste Sélectionner un attribut.

Sélectionnez une valeur correspondante dans la liste Sélectionner une valeur.

## Boîte de dialogue Filtrer les éléments par niveau

Cette zone de liste affiche les niveaux de hiérarchie disponibles dans le sous-ensemble en cours. Pour n'afficher que les éléments d'un certain niveau, sélectionnez ce niveau et cliquez sur OK.

Pour sélectionner plusieurs niveaux adjacents, cliquez et faites glisser la souris sur ces niveaux. Pour sélectionner plusieurs niveaux non adjacents, appuyez sur Ctrl et cliquez sur chaque niveau.





## Boîte de dialogue Filtrer un sous-ensemble

La boîte de dialogue Filtrer un sous-ensemble permet de créer un sous-ensemble d'après les valeurs d'un cube spécifié. Ainsi, vous pouvez créer un sous-ensemble de la dimension Région renvoyant les 10 éléments ayant la plus grande valeur pour les ventes annuelles réelles des véhicules Berline 1,8L dans le cube Ventés.

Cette boîte de dialogue contient les options suivantes.

Option	Description
NomCube	Le cube dont vous voulez filtrer les valeurs.
Filtre	<p>Le type de filtre à appliquer à la vue en cours.</p> <p><b>Nombres supérieurs</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les <math>n</math> éléments les plus grands, <math>n</math> étant un nombre spécifié dans l'option Valeur.</p> <p><b>Nombres inférieurs</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les <math>n</math> éléments les plus petits, <math>n</math> étant un nombre spécifié dans l'option Valeur.</p> <p><b>Somme supérieure</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les éléments les plus grands dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p> <p><b>Somme inférieure</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les éléments les plus petits dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p> <p><b>Pourcentage supérieur</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les éléments les plus grands dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un pourcentage du total de la dimension spécifié à l'aide de l'option Valeur.</p> <p><b>Pourcentage inférieur</b></p> <p>Filtre le sous-ensemble pour renvoyer uniquement les éléments les plus petits dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un pourcentage du total de la dimension spécifié à l'aide de l'option Valeur.</p> <p><b>Aucun</b></p> <p>Ne s'applique pas au filtrage des sous-ensembles.</p>
Valeur	Une valeur s'appliquant au type du Filtre.

Option	Description
Sélectionner des membres de colonne	Les éléments en colonne auxquels le filtre ou le tri est appliqué. Cliquez sur les boutons de dimension pour sélectionner un élément unique pour chaque dimension en colonne.
Trier	L'ordre de tri à appliquer aux éléments en colonne sélectionnés. <b>Croissant</b> Trie les valeurs des éléments en colonne spécifiés de la plus basse à la plus haute. <b>Décroissant</b> Trie les valeurs des éléments en colonne spécifiés de la plus haute à la plus basse. <b>Aucun</b> Pas d'ordre de tri.
Sélectionner des membres de colonne	Vous devez sélectionner un seul élément de chaque dimension de cube restante. Par exemple, si vous filtrez la dimension Région de la base de données d'exemple en utilisant les valeurs du cube Ventes, vous devez spécifier un seul élément de chacune des dimensions Modèle, Mois, RéceletBud et Compte1. Pour chaque dimension, cliquez sur le bouton approprié et sélectionnez un élément unique. Si le cube contient plus de 16 dimensions, cliquez sur  pour revenir à la page contenant les 16 dimensions précédentes ou sur  pour accéder à celle contenant les 16 dimensions suivantes.

## Boîte de dialogue Filtrer une vue

La boîte de dialogue Filtrer une vue permet de filtrer et trier des colonnes dans le Cube Viewer ou le Navigateur de feuille de calcul.

La boîte de dialogue contient les options suivantes.

Option	Filtre/Description
NomCube	Le cube dont vous voulez filtrer ou trier les valeurs. Cette option est toujours définie sur le cube associé à la vue en cours. Elle ne peut être modifiée.
Filtre	Le type de filtre à appliquer à la vue en cours.

Option	Filtre/Description
	<p>Nombres supérieurs</p> <p>Filtre la vue pour afficher uniquement les <math>n</math> éléments les plus grands, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p>
	<p>Nombres inférieurs</p> <p>Filtre la vue pour afficher uniquement les <math>n</math> éléments les plus petits, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p>
	<p>Somme supérieure</p> <p>Filtre la vue pour afficher uniquement les éléments les plus grands dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p>
	<p>Somme inférieure</p> <p>Filtre la vue pour afficher uniquement les éléments les plus petits dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un nombre spécifié à l'aide de l'option Valeur.</p>
	<p>Pourcentage supérieur</p> <p>Filtre la vue pour afficher uniquement les éléments les plus grands dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un pourcentage du total de la dimension spécifié à l'aide de l'option Valeur.</p>
	<p>Pourcentage inférieur</p> <p>Filtre la vue pour afficher uniquement les éléments les plus petits dont la somme est supérieure ou égale à <math>n</math>, <math>n</math> étant un pourcentage du total de la dimension spécifié à l'aide de l'option Valeur.</p>
	<p>Aucun</p> <p>Pas de filtre. Sélectionnez cette option pour trier les valeurs sans les filtrer.</p>
Valeur	Une valeur s'appliquant au type du Filtre.
Sélectionner des membres de colonne	Les éléments en colonne auxquels le filtre ou le tri est appliqué. Cliquez sur les boutons de dimension pour sélectionner un élément unique pour chaque dimension en colonne.
Trier	L'ordre de tri à appliquer aux éléments en colonne sélectionnés.

Option	Filtre/Description
	Croissant Trie les valeurs des éléments en colonne spécifiés de la plus basse à la plus haute.
	Décroissant Trie les valeurs des éléments en colonne spécifiés de la plus haute à la plus basse.
	Aucun Pas d'ordre de tri.

## Boîte de dialogue Choisir une vue (Navigateur de feuille de calcul)

La boîte de dialogue Choisir une vue permet d'ouvrir une vue sur le serveur local ou sur tout autre serveur disponible sur le réseau.

Zone	Description
Serveur	La liste Serveur affiche tous les serveurs disponibles sur votre réseau. Sélectionnez le serveur où réside la vue à ouvrir.  Si vous n'êtes pas connecté au serveur contenant cette vue, cliquez sur Connexion pour ouvrir la boîte de dialogue Connexion au serveur et vous connecter. Cliquez sur Démarrer le serveur local pour démarrer le serveur local.
Cube	La liste Cube affiche tous les cubes disponibles sur le serveur sélectionné. Sélectionnez le cube associé à la vue à ouvrir.
Vue	La liste Vue affiche toutes les vues disponibles sur le cube sélectionné. Sélectionnez la vue à ouvrir.

## Menu Navigateur de feuille de calcul

Le menu Navigateur de feuille de calcul est disponible si vous cliquez avec le bouton droit de la souris dans le contrôle de vue TM1. Ce menu permet d'ouvrir, mettre à jour, formater, faire des tranches ou enregistrer une vue. Il comprend aussi plusieurs options qui contrôlent les opérations du Navigateur de feuille de calcul.

Élément de menu	Description
Mettre la vue à jour	Met à jour la vue en cours en envoyant les valeurs modifiées à la base de données TM1 et en récupérant les valeurs actuelles de la base de données.
Choisir une vue	Ouvre la boîte de dialogue Choisir une vue qui permet d'ouvrir une vue sur n'importe quel serveur disponible.

<b>Élément de menu</b>	<b>Description</b>
Styles	Ouvre la boîte de dialogue Styles de vue qui permet de formater une vue.
Enregistrer	Ouvre la boîte de dialogue Enregistrer une vue qui permet d'enregistrer une vue TM1.
Effacer l'affichage	Efface toutes les données associées à une vue, y compris les étiquettes de titre, de ligne et de colonne.
Supprimer	Supprime le contrôle de vue TM1. Notez que toutes les données associées à la vue, y compris les étiquettes de titre, de ligne et de colonne, restent dans la feuille de calcul.
Couper	Coupe le contenu du contrôle de vue TM1 et le met dans le Presse-papiers.
Copier	Copie le contenu du contrôle de vue TM1 et le met dans le Presse-papiers.
Tranche	Crée une tranche de la vue en cours dans une nouvelle feuille de calcul Excel.
Supprimer les valeurs nulles	Ce bouton bascule supprime ou affiche les valeurs nulles dans la vue de cube. Une coche indique que les valeurs nulles sont supprimées dans la vue en cours.
Afficher automatiquement	Ce bouton-bascule active/désactive la mise à jour automatique de l'affichage lors de la reconfiguration de vue. Une coche indique que la vue est mise à jour automatiquement chaque fois que la configuration de vue change.
Mettre à jour après le recalcul	Ce bouton-bascule active ou désactive la mise à jour de vue automatique après le recalcul de la feuille de calcul (F9). Une coche indique que la vue est mise à jour automatiquement à chaque recalcul de la feuille de calcul.
Aide	Ouvre la rubrique d'aide Navigateur de feuille de calcul.

## Fenêtre Journal des messages

La fenêtre Journal des messages de TM1 affiche des messages de statut sur l'activité du serveur. Ces messages sont enregistrés dans le journal des messages du serveur et contiennent des informations détaillées sur l'activité telles que les processus exécutés, les tâches, les cubes et dimensions chargés et la réplication synchronisée.

Pour des informations détaillées sur le journal des messages du serveur, voir *TM1 Operations*.

### Journal des messages, sous-fenêtre

Cette sous-fenêtre affiche les messages de statut contenus dans le journal des messages du serveur.

Chaque ligne de la sous-fenêtre représente un message unique. Si un des messages du journal fait état d'une condition d'erreur pour un processus exécuté ou une réplication, vous pouvez double-cliquer dessus pour afficher les détails de la raison de l'erreur générée par l'activité.

Pour des détails sur les zones de la sous-fenêtre Journal des messages, voir *TM1 Operations*.

## Menu Fichier

Élément de menu	Description
Quitter	Ferme la fenêtre Journal des messages.

## Menu Edition

Élément de menu	Description
Copier	Copie le texte sélectionné dans la sous-fenêtre Journal des messages et le met dans le Presse-papiers.
Rechercher	Ouvre la boîte de dialogue Rechercher dans laquelle vous pouvez faire des recherches dans la sous-fenêtre journal des messages.

## Menu Aide

Élément de menu	Description
Aide du Journal des messages	Ouvre la rubrique d'aide Journal des messages.
Table des matières et Index	Ouvre la bibliothèque de documentation complète de TM1.

## Boîte de dialogue Nouvel attribut

Zone	Description
Nom du nouvel attribut	Entrez un nom pour le nouvel attribut dans cette zone.
Numérique	Sélectionnez cette option si les valeurs d'attribut sont des nombres.
Chaîne	Sélectionnez cette option si les valeurs d'attribut sont des chaînes de caractères.
Alias	Sélectionnez cette option si les valeurs d'attribut sont des alternatives pour l'élément, la dimension, le cube en cours ou des noms de serveur.

## Boîte de dialogue Ouvrir un sous-ensemble

Utilisez la boîte de dialogue Ouvrir un sous-ensemble pour ouvrir un sous-ensemble de dimension existant.

Pour ouvrir le sous-ensemble public par défaut, sélectionnez la case Par défaut et cliquez sur **Ouvrir**.

## Boîte de dialogue Ouvrir une vue

Utilisez la boîte de dialogue Ouvrir une vue pour ouvrir la vue d'un cube existant.

Pour ouvrir la vue publique par défaut, sélectionnez la case Par défaut et cliquez sur **Ouvrir**.

## Assistant d'impression de rapports

L'Assistant d'impression de rapports permet de générer des rapports de style "dossier de synthèse" à partir de tranches TM1.

L'Assistant est composé de trois écrans.

- Ecran 1 - Sélection des feuilles à inclure dans le rapport
- Ecran 2 - Sélection des dimensions en titre à utiliser dans le rapport, définition de l'ordre dans lequel elles apparaissent dans le rapport et définition des options d'impression du classeur.
- Ecran 3 - Sélection d'une destination d'impression pour le rapport (imprimante, fichier Excel ou PDF)

L'Assistant d'impression de rapports vous permet aussi d'enregistrer vos paramètres d'impression de rapports.

### Tous les écrans






Bouton	Description
Charger	Cliquez sur ce bouton pour charger un travail d'impression TM1 existant.
Enregistrer	Cliquez sur ce bouton pour enregistrer les paramètres de rapport actuels comme travail d'impression TM1.
Enregistrer sous	Cliquez sur ce bouton pour enregistrer les paramètres de rapport actuels comme travail d'impression TM1 sous un nouveau nom.
Suivant	Cliquez sur ce bouton pour avancer à l'écran de l'Assistant suivant.
Annuler	Cliquez sur ce bouton pour fermer la fenêtre de l'Assistant sans générer de rapport.

### Ecran 1 de 3



Élément	Description
Inclure ces feuilles dans la liste de rapports	Liste les feuilles de calcul disponibles dans le classeur en cours que vous pouvez inclure dans le rapport.  Pour inclure une feuille de calcul dans le rapport, cochez la case en face du nom de la feuille.

Élément	Description
Sélectionner tout	Cliquez sur ce bouton pour inclure toutes les feuilles dans le rapport.
Effacer tout	Cliquez sur ce bouton pour exclure toutes les feuilles du rapport.

### Ecran 2 de 3

Élément	Description
Liste des dimensions en titre disponibles	Liste les dimensions en titre disponibles que vous pouvez utiliser dans le rapport.  Pour chaque dimension, cette liste affiche le nom du sous-ensemble en cours (le cas échéant), le nombre d'éléments dans la dimension ou le sous-ensemble et l'adresse de cellule de la dimension en titre dans la feuille de calcul.
Liste des dimensions en titre sélectionnées	Liste les dimensions en titre à inclure dans le rapport.  L'ordre de cette liste est utilisé lorsque TM1 génère le rapport.
Ajouter 	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées de la liste Dimensions en titre disponibles à la liste Dimensions en titre sélectionnées.
Ajouter tout 	Cliquez sur ce bouton pour déplacer toutes les dimensions de la liste Dimensions en titre disponibles à la liste Dimensions en titre sélectionnées.
Supprimer 	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées de la liste Dimensions en titre sélectionnées à la liste Dimensions en titre disponible.
Supprimer tout 	Cliquez sur ce bouton pour déplacer toutes les dimensions de la liste Dimensions en titre sélectionnées à la liste Dimensions en titre disponibles.
Déplacer vers le haut 	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées vers le haut dans la liste Dimensions en titre sélectionnées. L'ordre de cette liste est utilisé lorsque TM1 génère le rapport.



Elément	Description
Déplacer vers le bas 	Cliquez sur ce bouton pour déplacer les dimensions sélectionnées vers le bas dans la liste Dimensions en titre sélectionnées. L'ordre de cette liste est utilisé lorsque TM1 génère le rapport.
Editeur de sous-ensemble 	Cliquez sur ce bouton pour ouvrir l'Editeur de sous-ensemble si vous souhaitez sélectionner un sous-ensemble d'éléments de la dimension actuellement sélectionnée dans la liste Dimensions en titre sélectionnées.
Imprimer un seul classeur	Sélectionnez cette option pour créer un rapport organisé en un groupe de feuilles de calcul complet.  Chaque feuille du rapport n'est imprimée qu'une fois, y compris les feuilles utilisateur ne contenant pas de données de tranche TM1.
Imprimer plusieurs classeurs	Sélectionnez cette option pour créer un rapport organisé en plusieurs groupes en fonction des éléments de dimension.  Cette option crée un rapport comportant un grand nombre de feuille, car une copie de chaque feuille est imprimée pour chaque élément en titre.
Nombre total de classeurs Excel qui seront générés	Affiche le nombre total de feuilles Excel que TM1 générera pour le rapport en cours.

### Ecran 3 de 3

Zone	Description
Imprimer sur une imprimante	Sélectionnez cette option si vous voulez imprimer un rapport avec une imprimante.
Enregistrer comme fichiers Excel	Sélectionnez cette option si vous voulez générer le rapport sous la forme de fichier Excel.
Enregistrer comme fichiers PDF	Sélectionnez cette option si vous voulez générer le rapport sous la forme de fichier PDF.
Aperçu	Ce bouton devient disponible quand vous sélectionnez l'option Imprimer sur une imprimante.  Cliquez sur ce bouton pour afficher un aperçu du rapport afin de l'imprimer.

Zone	Description
Nom de l'imprimante	<p>Ce bouton devient disponible quand vous sélectionnez l'option Imprimer sur une imprimante.</p> <p>Utilisez cette option pour indiquer l'imprimante sur laquelle TM1 imprime le rapport.</p>
Nombre de copies	<p>Ce bouton devient disponible quand vous sélectionnez l'option Imprimer sur une imprimante.</p> <p>Utilisez cette option pour indiquer le nombre de copies du rapport à imprimer.</p>
Imprimer dans un fichier	<p>Ce bouton devient disponible quand vous sélectionnez l'option Imprimer sur une imprimante.</p> <p>Sélectionnez cette option pour enregistrer ce rapport comme rapport prêt pour l'impression.</p>
Nom du fichier	<p>Ce bouton devient disponible quand vous sélectionnez les deux options Imprimer sur une imprimante et Imprimer dans un fichier.</p> <p>Entrez un chemin complet et un nom de fichier pour le rapport que vous allez imprimer. Vous devez également indiquer un type de fichier. Par exemple, si vous imprimez dans un fichier en utilisant une imprimante PostScript, vous devez ajouter l'extension .ps au nom du fichier.</p>
Parcourir	<p>Ce bouton devient disponible quand vous sélectionnez l'option d'imprimer ou d'enregistrer le rapport dans un fichier.</p> <p>Cliquez sur ce bouton pour sélectionner le répertoire dans lequel vous souhaitez enregistrer le rapport.</p>
Assembler	<p>Ce bouton devient disponible quand vous sélectionnez l'option Imprimer sur une imprimante.</p> <p>Sélectionnez cette option pour collationner les pages quand vous imprimez plusieurs copies d'un rapport.</p>
Générer un nouveau classeur pour chaque titre	<p>Cette option devient disponible quand vous choisissez d'enregistrer le rapport comme fichier Excel ou PDF.</p> <p>Sélectionnez cette option si vous souhaitez créer un fichier distinct pour chaque dimension en titre du rapport.</p>

Zone	Description
Nom du répertoire	Cette option est disponible quand vous enregistrez un rapport comme fichier Excel ou PDF et que vous sélectionnez l'option Générer un nouveau classeur pour chaque titre.  Entrez un répertoire dans lequel enregistrer les fichiers de rapport. Pour choisir un emplacement de répertoire, cliquez sur le bouton Parcourir.
Créer un instantané	Cette option devient disponible quand vous sélectionnez l'option Enregistrer comme fichiers Excel.  Sélectionnez cette option quand vous souhaitez enregistrer le rapport en format de fichier Excel contenant des valeurs et non pas des fonctions TM1 extrayant des valeurs.
Précédent	Cliquez sur ce bouton pour revenir à l'écran de l'Assistant précédent.
Terminer	Cliquez sur ce bouton pour enregistrer le rapport sur la base des options que vous avez sélectionnées.

## Boîte de dialogue Options de processus

La boîte de dialogue Options du processus contrôle le comportement du bouton d'action *avant* et *après* l'exécution du processus.

Utilisez une des méthodes suivantes pour définir le texte des messages de confirmation et de statut qui s'affichent lorsque vous cliquez sur le bouton d'action.

- Entrez le texte du message directement dans la zone de message.
- Servez-vous d'une référence Excel pour extraire dynamiquement le texte d'un message d'une feuille de calcul.

Par exemple, pour extraire le texte d'un message depuis le contenu de la cellule A1, entrez =A1 dans la zone de texte pour ce message. Utilisez le format ci-dessous pour référencer une plage nommée : =*Plage nommée*.

Pour plus d'informations sur l'utilisation de la boîte de dialogue Options de processus, voir IBM Cognos *TM1 for Developers*.

Zone	Description
Recalculer le classeur automatiquement	Sélectionnez cette option pour que TM1 recalcule automatiquement la totalité du classeur après l'exécution du processus.
Afficher un message de réussite	Sélectionnez cette option pour afficher un message <i>après</i> que le processus ait été exécuté avec succès.  Entrez le texte de votre message dans la zone, comme décrit ci-dessus.

Zone	Description
Afficher un message d'échec	Sélectionnez cette option pour afficher un message si le processus n'a pas pu être exécuté avec succès.  Entrez le texte de votre message dans la zone, comme décrit ci-dessus.
Afficher la boîte de dialogue de confirmation	Sélectionnez cette option pour afficher une zone de message de confirmation (Oui/Non) <i>avant</i> le lancement du processus. L'utilisateur peut cliquer sur Oui pour exécuter le processus ou sur Non pour l'annuler.  Entrez le texte de votre message dans la zone, comme décrit ci-dessus.
OK	Cliquez sur ce bouton pour enregistrer vos réglages et fermer la boîte de dialogue.
Annuler	Cliquez sur ce bouton pour fermer la boîte de dialogue sans enregistrer vos réglages.

## Boîte de dialogue Répliquer un cube

Utilisez la boîte de dialogue Répliquer un cube pour répliquer un cube d'un serveur source à un serveur cible.

### Informations sur le cube

Élément	Description
Nom	Le nom du cube miroir sur le serveur cible.  Par défaut, TM1 nomme le cube miroir en concaténant le nom du serveur source et le nom du cube source.  Ne changez pas le nom par défaut si vous répliquez les règles de ce cube.
Copier les données et régler sur Synchroniser	Sélectionnez cette option pour copier des données quand la réplication est établie et pour synchroniser les données quand la synchronisation a lieu entre le serveur source et le serveur cible.
Copier les données mais ne pas régler sur Synchroniser	Sélectionnez cette option pour copier les données lorsque la synchronisation est établie, mais pour désactiver la synchronisation ultérieure des données.
Répliquer les vues	Sélectionnez cette option pour répliquer toutes les vues associées au cube source.

## Informations sur les règles

Élément	Description
Copier la règle	Sélectionnez cette option pour copier les règles du cube source au cube miroir.
Régler la règle à synchroniser	Activez cette case pour synchroniser les règles quand la synchronisation a lieu entre les serveurs source et cible.  Désactivez cette case pour désactiver la synchronisation de la règle.
Ne pas copier la règle	Si vous sélectionnez cette option, TM1 ne copie pas la règle du cube source vers le cube miroir.

## Informations sur les dimensions

Élément	Description
Zone Informations sur les dimensions	Cette zone affiche des informations sur les dimensions du cube miroir.  Si le cube source ne contient pas de règles, TM1 renomme les dimensions miroir en concaténant les noms de serveur source et les noms de dimension source.  Si le cube source contient des règles, TM1 ne change pas les noms de dimension dans le cube miroir.  La zone Informations sur les dimensions affiche également le nom de la dimension source, du serveur source, et le statut de réplication pour chaque dimension du cube.
Sélectionner une dimension locale	Pour utiliser une dimension locale au lieu d'une dimension source, cliquez sur la dimension source dans la zone Informations sur les dimensions, puis sur Sélectionner une dimension locale. Sélectionnez la dimension locale à utiliser et cliquez sur OK.
Rétablir la sélection en cours aux valeurs par défaut	Si vous changez des options d'informations de dimension pour une dimension d'un cube répliqué, vous pouvez restaurer toutes les options aux valeurs par défaut en sélectionnant la dimension dans la case Informations sur la dimension et en cliquant sur ce bouton.

Élément	Description
Remplacer une dimension	Cette option est disponible quand vous sélectionnez une dimension locale. Sélectionnez cette option pour remplacer la dimension locale par la définition de la dimension source.
Régler la dimension à synchroniser	Remplissez cette case pour synchroniser les changements entre la dimension source et la dimension miroir quand la synchronisation a lieu entre les serveurs source et cible. Désactivez cette case pour désactiver la synchronisation de la dimension.
Ne pas remplacer la dimension	Cette option est disponible quand vous sélectionnez une dimension locale. Sélectionnez cette option pour utiliser la dimension locale telle quelle.
Répliquer des sous-ensembles	Sélectionnez cette option pour répliquer tous les sous-ensembles associés à la dimension source.

## Editeur de règles

L'Editeur de règles propose un ensemble complet de menus permettant de créer, modifier et gérer les règles TM1. Des raccourcis claviers sont fournis pour les options de menu utilisées le plus souvent.

### Menu Fichier

Le tableau ci-dessous décrit toutes les options du menu Fichier.

Nom	Description
Importer	Ouvre une boîte de dialogue de navigation de fichiers pour vous permettre de sélectionner un fichier texte à importer. Le texte importé remplace, le cas échéant, la règle en cours.
Enregistrer	Enregistre la règle en cours sur le serveur.
Enregistrer sous	Enregistre la règle en cours dans un fichier .rux de règle TM1 externe.
Vérifier la syntaxe	Vérifie que la règle en cours ne contient pas d'erreurs de syntaxe.
Imprimer	Ouvre la boîte de dialogue Imprimer à l'aide de laquelle vous pouvez imprimer la règle en cours.

Nom	Description
Aperçu avant impression	Ouvre la fenêtre Aperçu avant impression dans laquelle vous pouvez consulter une version imprimée d'exemple de la règle avant de l'envoyer à l'impression.
Quitter	Ferme l'Editeur de règles.

## Menu Edition

Le tableau ci-dessous décrit toutes les options du menu Edition.

Nom	Description
Annuler	Annule la dernière édition. Le programme prend en charge plusieurs niveaux d'annulation.
Rétablir	Annule la dernière commande "Annuler".
Couper	Supprime le texte sélectionné et le met dans le Presse-papiers
Copier	Copie le texte sélectionné et le met dans le Presse-papiers.
Coller	Colle le contenu du Presse-papiers dans l'Editeur de règles.
Sélectionner tout	Sélectionne tout le contenu de l'Editeur de règles.
Rechercher	Ouvre la boîte de dialogue Rechercher dans laquelle vous pouvez faire des recherches de texte dans la règle.
Rechercher/Remplacer	Ouvre la boîte de dialogue Rechercher/Remplacer dans laquelle vous pouvez rechercher du texte et le remplacer.
Suivant	Identifie l'occurrence suivante du texte que vous recherchez.
Basculer le signet	Active ou désactive un signet pour la ligne de code en cours.
Signet suivant	Déplace le curseur au signet disponible suivant.
Signet précédent	Déplace le curseur au signet disponible précédent.
Effacer tous les signets	Supprime tous les signets.

Nom	Description
Commenter la sélection	Ajoute un symbole de commentaire (#) sur toutes les lignes du texte sélectionné pour exclure ces lignes de la règle compilée.  <b>Remarque :</b> La longueur du commentaire est limitée à 255 octets. Pour les jeux de caractères occidentaux, tels que l'anglais ou le français, un caractère est représenté par un seul octet, ce qui vous permet d'entrer des commentaires de 255 caractères. Cependant, les jeux de caractères de grande taille, tels que le chinois, le japonais et le coréen, utilisent plusieurs octets pour représenter un caractère. Dans ce cas, la limite de 255 octets peut être dépassée plus tôt, et donc ne pas permettre la saisie de 255 caractères.
Décommenter la sélection	Supprime symbole de commentaire (#) de toutes les lignes du texte sélectionné pour inclure ces lignes de la règle compilée.
Mettre en retrait	Met en retrait les lignes actuellement sélectionnées.
Supprimer le retrait	Supprime le retrait des lignes actuellement sélectionnées.
Aller à la ligne...	Affiche la boîte de dialogue Aller à la ligne, qui vous permet de saisir un numéro de ligne dans l'Editeur de règles et d'y aller directement

## Menu Affichage

Le tableau ci-dessous décrit toutes les options du menu Affichage.

**Remarque :** Les modifications que vous effectuez dans les réglages du menu Affichage sont enregistrées quand vous quittez l'Editeur de règles et sont réappliquées à la prochaine ouverture de l'Editeur de règles.

Nom	Description
Retour à la ligne	Active ou désactive la fonction de retour à la ligne : les lignes de texte s'étendent alors vers la droite ou vont à la ligne lorsqu'elles sont affichées dans la sous-fenêtre Edition.
Numéros de ligne	Active ou désactive les numéros de ligne.
Infobulles de fonctions	Active ou désactive l'affichage des infobulles de fonctions.
Saisie semi-automatique	Active ou désactive la fonction de saisie automatique quand vous saisissez du texte dans la sous-fenêtre Edition.



Nom	Description
Barre d'outils	Active ou désactive l'affichage de la barre d'outils principale.
Barre de statut	Active ou désactive l'affichage de la barre de statut au bas de l'Editeur de règles.
Objets de contrôle	Active ou désactive l'affichage des objets de contrôle TM1 lorsque vous sélectionnez des cubes.
Développer toutes les régions	Développe toutes les régions définies par l'utilisateur dans la règle en cours pour montrer toutes les lignes.
Réduire toutes les régions	Réduit toutes les régions définies par l'utilisateur dans la règle en cours pour masquer toutes les lignes comprises dans une région.

## Menu Insertion

Le tableau ci-dessous décrit toutes les options du menu Insertion.

Nom	Description
Fonction	Affiche la boîte de dialogue Insérer une fonction qui permet d'entrer une nouvelle fonction dans la règle en cours.
Référence de cube	Affiche la boîte de dialogue Insérer une référence de cube pour vous permettre d'insérer une fonction SGBD.

## Menu Outils

Le tableau ci-dessous décrit toutes les options du menu Outils.

Nom	Description
Préférences...	Affiche la boîte de dialogue Préférences dans laquelle vous pouvez définir les attributs de police tels que le type de police, la taille et la couleur à utiliser dans la sous-fenêtre Edition.
Options...	Affiche la boîte de dialogue Option de contrôle dans laquelle vous pouvez modifier les paramètres globaux de l'Editeur de règles.

## Boîte de dialogue Enregistrer le sous-ensemble

---

<b>Zone</b>	<b>Description</b>
Sélectionner ou saisir un nom de sous-ensemble	Entrez un nom pour le sous-ensemble enregistré ou sélectionnez un nom existant dans la liste.
Privé	Basculez cette option pour enregistrer le sous-ensemble comme objet privé. Basculez cette option pour enregistrer le sous-ensemble comme objet public.
Par défaut	Basculez cette option pour enregistrer le sous-ensemble comme sous-ensemble par défaut.
Enregistrer une expression	Si le sous-ensemble est dynamique, basculez cette option pour enregistrer l'expression MDX avec le sous-ensemble.  Si le sous-ensemble est dynamique et que vous ne basculez pas l'option, l'expression MDX n'est pas enregistrée. Le sous-ensemble qui en résulte est statique et contient les éléments présents lors de l'enregistrement.

## Boîte de dialogue Enregistrer la vue

---

<b>Zone</b>	<b>Description</b>
Sélectionner ou saisir une vue nommée	Entrez un nom pour la vue enregistrée ou sélectionnez un nom existant dans la liste.
Privé	Basculez cette option pour enregistrer la vue comme objet privé. Basculez cette option pour enregistrer la vue comme objet public.
Par défaut	Basculez cette option pour enregistrer la vue comme vue par défaut.

## Boîte de dialogue Enregistrer la vue (Navigateur de feuille de calcul)

---

<b>Zone</b>	<b>Description</b>
Nom de la vue	Entrez un nom pour la vue dans cette zone.
Privé	Basculez cette option pour enregistrer la vue comme objet privé. Basculez cette option pour enregistrer la vue comme objet public.
Par défaut	Basculez cette option pour enregistrer la vue comme vue par défaut.

## Boîte de dialogue Affectations de sécurité

La boîte de dialogue Affectations de sécurité permet d'affecter les privilèges d'accès pour les cubes, les structures de dimension, les éléments individuels et les tâches. Les privilèges d'accès sont affectés par groupe d'utilisateurs.

### Grille Affectations

La grille Affectations affiche le nom des objets comme en-têtes de ligne et celui des groupes d'utilisateurs comme en-têtes de colonne. Les privilèges d'accès apparaissent comme des valeurs de cellule à l'intersection d'un objet donné et d'un groupe d'utilisateurs.

Quand vous accédez à la boîte de dialogue Affectation de sécurité à partir d'un groupe Cubes, la grille comprend une colonne Journalisation. Cette colonne comprend une case à cocher pour chaque cube. Pour activer la journalisation pour un cube, cliquez sur la case à cocher à l'intersection du nom du cube et de la colonne Journalisation. Pour désactiver la journalisation, désactivez cette case à cocher. La valeur par défaut est option activée.

### Privilèges d'accès

Cliquez sur une des options suivantes pour affecter un privilège d'accès à une cellule sélectionnée dans la Grille d'affectations.

#### Aucun privilège

Le tableau suivant décrit les possibilités d'accès de groupes d'utilisateurs TM1 à divers objets TM1 si aucun privilège ne leur a été affecté pour un objet.

Objet	Description
Cube	Les membres du groupe ne peuvent pas voir le cube dans l'Explorateur de serveur et ne peuvent donc pas le parcourir.
Élément	Les membres du groupe ne peuvent ni afficher l'élément dans l'Editeur de sous-ensemble ou l'Editeur de dimension, ni visualiser les cellules identifiées par l'élément lors de la navigation dans le cube.
Dimension	Les membres du groupe ne peuvent ni afficher la dimension dans l'Explorateur de serveur, ni parcourir les cubes contenant cette dimension.
Processus	Les membres du groupe ne peuvent pas voir le processus dans l'Explorateur de serveur. <b>Remarque :</b> Les privilèges affectés aux processus sont ignorés lorsqu'un processus est exécuté depuis une tâche.
Tâche	Les membres du groupe ne peuvent pas voir la tâche dans l'Explorateur de serveur.
Application	Les membres du groupe ne peuvent voir ni l'application, ni son contenu dans l'Explorateur de serveur.

Objet	Description
Référence	Les membres du groupe ne peuvent pas voir la référence dans l'Explorateur de serveur.

### Privilège de lecture

Le tableau suivant décrit les possibilités d'accès de groupes d'utilisateurs TM1 à divers objets TM1 si le privilège de lecture leur a été affecté pour un objet.

Objet	Description
Cube	Les membres du groupe peuvent voir les données du cube, mais ne peuvent pas les modifier.
Élément	Les membres du groupe peuvent voir les données identifiées par l'élément, mais ne peuvent pas les modifier.
Dimension	Les membres du groupe peuvent visualiser les éléments d'une dimension, mais ne peuvent pas modifier la structure de cette dimension.
Processus	Les membres du groupe peuvent visualiser le processus dans l'Explorateur de Serveur et l'exécuter, mais pas le modifier.  <b>Remarque :</b> Les privilèges affectés aux processus sont ignorés lorsqu'un processus est exécuté depuis une tâche.
Tâche	Les membres du groupe peuvent visualiser la tâche dans l'Explorateur de Serveur et l'exécuter manuellement, mais ne peuvent pas la modifier, ni changer le statut d'activation.
Application	Les membres du groupe peuvent voir l'application et utiliser des références dans l'application pour laquelle le groupe a au moins le privilège Lecture. Vous pouvez créer des références privées dans l'application, ainsi que des sous-applications privées.
Référence	Les membres du groupe peuvent ouvrir et utiliser la référence, mais ne peuvent pas la mettre à jour dans l'application parente. Vous pouvez toutefois réaliser une opération "enregistrer sous" pour enregistrer une nouvelle session privée de la référence dans toute application pour laquelle vous avez au moins le privilège Lecture.

### Privilège d'écriture

Le tableau suivant décrit les possibilités d'accès de groupes d'utilisateurs TM1 à divers objets TM1 si le privilège d'écriture leur a été affecté pour un objet.

Objet	Description
Cube	Les membres du groupe peuvent visualiser et modifier les données du cube et créer des vues privées pour ce cube.  L'accès en écriture ne vous permet pas de modifier les données identifiées par des éléments consolidés ou dérivées de règles. Par définition, les valeurs dérivées d'une consolidation ou de règles ne peuvent pas être modifiées.
Élément	Les membres du groupe peuvent visualiser et modifier les données identifiées par l'élément.
Dimension	Chaque membre du groupe peut modifier les attributs d'éléments et les formats d'éléments et créer des sous-ensembles privés pour la dimension. Les membres du groupe peuvent également modifier les attributs de la dimension elle-même.

### Privilège de réservation

Le tableau suivant décrit les possibilités d'accès de groupes d'utilisateurs TM1 à divers objets TM1 si le privilège de réservation leur a été affecté pour un objet.

Notez que quand vous réservez un objet, la réservation expire quand le serveur sur lequel réside l'objet est arrêté.

Objet	Description
Cube	Les membres du groupe peuvent visualiser et modifier les données de cube, ainsi que le réserver pour empêcher d'autres clients d'en modifier les données. Vous pouvez libérer un cube que vous avez réservé.
Élément	Les membres du groupe peuvent visualiser et modifier les données identifiées par l'élément, ainsi que le réserver pour empêcher les autres utilisateurs d'en modifier les données. Vous pouvez libérer un élément que vous avez réservé.
Dimension	Les membres du groupe peuvent ajouter, supprimer et réorganiser les éléments de la dimension, et réserver celle-ci pour empêcher d'autres utilisateurs d'en modifier la structure. Vous pouvez libérer une dimension que vous avez réservée.

### Privilège de verrouillage

Le tableau suivant décrit les possibilités d'accès de groupes d'utilisateurs TM1 à divers objets TM1 si le privilège de verrouillage leur a été affecté pour un objet.

Notez qu'il n'existe pas de privilège de déverrouillage et que seuls les utilisateurs ayant le privilège d'administration pour un objet peuvent déverrouiller cet objet.

<b>Objet</b>	<b>Description</b>
Cube	Les membres du groupe peuvent visualiser et modifier les données du cube, et aussi le verrouiller.  Quand un cube est verrouillé, personne ne peut mettre à jour ses données.
Élément	Les membres du groupe peuvent visualiser et modifier les données identifiées par l'élément, et aussi le verrouiller.  Quand un élément est verrouillé, personne ne peut mettre à jour les données identifiées par cet élément.
Dimension	Les membres du groupe peuvent ajouter, supprimer et réorganiser les éléments de la dimension, et verrouiller celle-ci pour empêcher d'autres utilisateurs d'en modifier la structure.  Quand une dimension est verrouillée, personne ne peut en modifier la structure.

#### **Privilège d'administration**

Le tableau suivant décrit les possibilités d'accès d'un groupe d'utilisateurs TM1 à différents objets TM1 sur lesquels ils ont reçu le privilège Admin.

<b>Objet</b>	<b>Description</b>
Cube	Les membres du groupe peuvent lire, écrire, réserver, libérer, verrouiller, déverrouiller et supprimer le cube.
Élément	Les membres du groupe peuvent visualiser, mettre à jour et supprimer les cellules identifiées par l'élément. Ils peuvent réserver, libérer, verrouiller et déverrouiller l'élément. Ils peuvent aussi accorder des privilèges d'accès à cet élément à d'autres utilisateurs.
Dimension	Les membres du groupe peuvent ajouter, supprimer ou réorganiser les éléments de la dimension. Ils peuvent réserver, libérer, verrouiller et déverrouiller la dimension. Ils peuvent aussi créer des sous-ensembles publics pour la dimension et accorder des privilèges d'accès pour la dimension à d'autres utilisateurs.

Objet	Description
Application	<p>Les membres du groupe peuvent visualiser l'application, utiliser les références se trouvant dans l'application et créer des références publiques et privées dans l'application. Ils peuvent également créer des sous-applications publiques et privées.</p> <p>Quand un groupe a le privilège Admin pour une application, les membres de ce groupe peuvent définir les privilèges de sécurité pour toutes les références et sous-applications se trouvant dans l'application des autres groupes, mais pas celle de leur propre groupe.</p>
Référence	<p>Les membres du groupe peuvent utiliser la référence ainsi que la mettre à jour ou la supprimer. Ils peuvent publier les références privées et privatiser les références publiques.</p>

### Sélectionner une dimension

Quand vous accédez à la boîte de dialogue Affectation de sécurité depuis une dimension individuelle, l'option Sélectionner une dimension devient disponible. Cette option permet d'affecter les privilèges d'accès aux éléments de plusieurs dimensions.

Après avoir affecté les privilèges d'accès pour une dimension, cliquez sur Enregistrer pour sélectionner une nouvelle dimension dans la liste Sélectionner une dimension. Quand vous avez fini d'affecter tous les privilèges d'accès aux dimensions voulues, cliquez sur OK pour fermer la boîte de dialogue.

### Boîte de dialogue Sélectionner un cube

Sélectionnez le nom du cube à insérer dans la feuille de calcul ou la formule et cliquez sur **OK**.

### Boîte de dialogue Sélectionner un cube pour les règles

Sélectionnez le cube pour lequel vous voulez créer une règle et cliquez sur **OK**.

### Boîte de dialogue Sélectionner une dimension

Sélectionnez le nom de la dimension à insérer dans la feuille de calcul ou la formule et cliquez sur **OK**.

### Boîte de dialogue Sélectionner un élément

Sélectionnez le nom de l'élément à insérer dans la feuille de calcul ou la formule et cliquez sur **OK**.

### Explorateur de serveur (Fenêtre principale)

### Sous-fenêtre de gauche (arborescence)

Affiche une représentation hiérarchique de tous les objets se trouvant sur le serveur auquel vous êtes connecté.

### Sous-fenêtre de droite (Propriétés)

Affiche les propriétés de l'objet sélectionné dans la sous-fenêtre gauche de l'Explorateur de serveur. Les propriétés varient selon l'objet sélectionné.

## Menu Fichier

Les options ci-après sont disponibles dans le menu Fichier de l'Explorateur de serveur.

Élément de menu	Description
Options	Ouvre la boîte de dialogue Options TM1.
Arrêter le serveur local	Arrête le serveur local et vous invite à enregistrer les modifications apportées aux données. Cette option n'est disponible que si le serveur local est actif.
Lancer le serveur local	Lance le serveur local. Cette option n'est disponible que si le serveur local est inactif.
Rafraîchir les serveurs disponibles	Met à jour l'affichage des serveurs disponibles dans la sous-fenêtre gauche de l'Explorateur de serveur.
Quitter	Ferme l'Explorateur de serveur et toutes les autres fenêtres associées à TM1 Perspectives/TM1 Architect.

## Menu dynamique

Les options disponibles dans le deuxième menu de l'Explorateur de serveur varient selon le type de l'objet actuellement sélectionné.

### Groupe de serveurs

Les options suivantes sont disponibles dans le menu TM1 lorsque vous sélectionnez le groupe de serveurs dans l'Explorateur de serveur.

Option	Description
Enregistrer toutes les données	Enregistre les données sur tous les serveurs auxquels vous êtes actuellement connecté.

### Serveur

Les options suivantes sont disponibles dans le menu Serveur quand vous sélectionnez un serveur précis dans l'Explorateur de serveur.

Option	Description
Enregistrer les données	Enregistre toutes les modifications des données sur le serveur sélectionné.



Option	Description
Recycler (Remettre à zéro la mémoire du serveur local)	Arrête et relance le serveur local. Quand vous utilisez cette option, vous avez le choix entre recycler en enregistrant les données sur le serveur local et recycler en abandonnant les modifications sur le serveur local.
Arrêter	Arrête le serveur local. Cette option n'est disponible que si le serveur local est sélectionné.
Sécurité, Clients/Groupes	Ouvre l'Editeur de clients/groupes pour le serveur sélectionné. Vous devez disposer des privilèges Admin pour le serveur pour pouvoir accéder à l'Editeur de clients/groupes.
Sécurité, Changer de mot de passe	Ouvre la boîte de dialogue Changer le mot de passe qui vous permet de changer de mot de passe sur le serveur sélectionné.
Sécurité, Rafraîchir la sécurité	Met à jour toutes les structures/affectations de sécurité sur le serveur sélectionné.

Option	Description
Affectation de droits	<p>Permet aux administrateurs de définir des droits d'accès à certaines fonctions pour des groupes d'utilisateurs individuels. A l'intersection du groupe d'utilisateurs et de la fonction, les administrateurs ont la possibilité de choisir Accorder ou Refuser (ou vide) pour activer ou désactiver la fonction en question. Il se peut que certaines fonctions activées soient ignorées, selon les paramètres de configuration définis sur le serveur.</p> <p>Les capacités suivantes peuvent être accordées à un groupe d'utilisateurs :</p> <ul style="list-style-type: none"> <li>• <b>Bloquer l'accès à l'explorateur de serveur</b> <p>Pour empêcher le lancement de l'Explorateur de serveur, cliquez sur l'intersection de cette capacité et du groupe d'utilisateurs, et sélectionnez <b>Accorder</b>. Si vous choisissez Refuser ou que vous ne renseignez aucune valeur, l'Explorateur de serveur est utilisé par ce groupe d'utilisateurs.</p> </li> <li>• <b>Mode d'écriture différée de l'espace de travail personnel</b> <p>Pour autoriser un groupe d'utilisateurs à utiliser des espaces de travail personnels, cliquez sur l'intersection du groupe d'utilisateurs et de cette capacité et sélectionnez <b>Accorder</b>. Si vous choisissez Refuser ou que vous ne renseignez aucune valeur, le groupe d'utilisateurs concerné n'utilisera pas les espaces de travail personnels.</p> <p>Si DisableSandboxing a la valeur T, cette affectation est ignorée.</p> </li> <li>• <b>Bac à sable</b> <p>Pour autoriser un groupe d'utilisateurs à utiliser des bacs à sable pour créer plusieurs scénarios de simulation, cliquez sur l'intersection de cette capacité et du groupe d'utilisateurs, et sélectionnez <b>Accorder</b>. Si vous choisissez Refuser ou que vous ne renseignez aucune valeur, le groupe d'utilisateurs concerné ne peut pas utiliser plusieurs bacs à sable.</p> <p>Si DisableSandboxing a la valeur T, cette affectation est ignorée.</p> </li> </ul> <p>Pour plus d'informations, reportez-vous à la documentation <i>TM1 Operations</i> et <i>TM1 Architect, Perspectives, and TM1 Web</i>.</p>
Afficher le journal des transactions	Ouvre la boîte de dialogue Requête du journal des transactions dans laquelle vous pouvez visualiser un journal des transactions sur le serveur sélectionné.

<b>Option</b>	<b>Description</b>
Afficher le journal des messages	Ouvre la boîte de dialogue Journal des messages qui affiche les messages enregistrés sur le serveur sélectionné.
Lancer le superviseur de performance	Lance la surveillance des performances. Lorsque le superviseur de performance fonctionne, TM1 alimente plusieurs cubes de contrôle permettant un suivi des statistiques des cubes, des clients et du serveur.
Arrêter le superviseur de performance	Arrête la surveillance des performances.
Mises à jour différées, Démarrer les mises à jour par lots	Lance les mises à jour par lots à envoyer au serveur sélectionné.
Mises à jour différées, Terminer les mises à jour par lots	Met fin aux mises à jour par lots et envoie toutes les modifications au serveur sélectionné.
Gestionnaire de serveurs	Ouvre la boîte de dialogue Centre de messagerie clients qui permet d'arrêter le serveur sélectionné, de déconnecter des clients et de diffuser des messages.
Annuler l'arrêt	Annule un arrêt de serveur déjà exécuté.
Se déconnecter	Déconnecte votre client du serveur sélectionné.
Qui suis-je	Renvoie un message donnant votre nom d'utilisateur sur le serveur.

### **Applications**

Les options suivantes sont disponibles dans le menu Applications quand vous sélectionnez soit le groupe Applications, soit une application individuelle dans l'Explorateur de serveur.

<b>Option</b>	<b>Description</b>
Ouvrir	Développe l'application ou le groupe d'applications sélectionné pour révéler les références et les sous-applications.
Fermer	Réduit l'application ou le groupe d'applications sélectionné pour masquer les références et les sous-applications.
Supprimer	Supprime l'application sélectionnée. Lorsque vous supprimez une application, toutes les sous-applications et les références dans l'application sont supprimées automatiquement. Cette option n'est pas disponible quand le groupe Applications est sélectionné.

Option	Description
Renommer	Définit le nom de l'application sélectionnée en mode d'édition pour vous permettre d'entrer un nouveau nom pour l'application. Cette option n'est pas disponible quand le groupe Applications est sélectionné.
Sécurité, Affectations de sécurité	Ouvre la boîte de dialogue Affectations de sécurité de TM1, qui permet d'affecter les privilèges de sécurité des références et sous-applications immédiates dans l'application ou groupe d'applications sélectionné.
Sécurité, Rendre publique	Sélectionnez cette option pour publier une application privée. Lorsque vous publiez une application, toutes les sous-applications et les références privées à des objets publics de l'application sont aussi publiées automatiquement. Cette option n'est pas disponible quand le groupe Applications est sélectionné.
Sécurité, Rendre privée	Sélectionnez cette option pour privatiser une application publique. Lorsque vous privatisez une application, toutes les sous-applications et les références publique dans l'application sont aussi privatisées automatiquement. Cette option n'est pas disponible quand le groupe Applications est sélectionné.

### Cubes

Les options suivantes sont disponibles dans le menu Cubes quand vous sélectionnez un groupe de cubes dans l'Explorateur de serveur.

Option	Description
Créer un nouveau cube	Ouvre la boîte de dialogue Création de cube.
Editer les attributs	Ouvre l'Editeur d'attributs pour le cube sélectionné.
Affectations de sécurité	Ouvre la boîte de dialogue Affectations de sécurité de TM1 pour les cubes du groupe de cubes sélectionné. Vous devez appartenir au groupe Admin pour le serveur contenant ce groupe de cubes pour accéder à la boîte de dialogue.

### Cube

Les options suivantes sont disponibles dans le menu Cube quand vous sélectionnez un cube dans l'Explorateur de serveur.

Option	Description
Parcourir	Ouvre le cube à parcourir dans la fenêtre du Cube Viewer.

Option	Description
Parcourir dans Excel	Ouvre le cube à parcourir dans le Navigateur de feuille de calcul.
Choisir	Copie le nom du cube dans le Presse-papiers.
Créer un nouveau cube	Ouvre la boîte de dialogue Création de cube.
Décharger un cube	Décharge le cube sélectionné de la mémoire du serveur.
Supprimer un cube	Supprime le cube sélectionné et toutes les données associées. Vous devez avoir le privilège Admin pour pouvoir supprimer un cube.
Réorganiser les dimensions	Ouvre la fenêtre Optimiseur de cube, dans laquelle vous pouvez optimiser l'ordre des dimensions dans le cube sélectionné.
Créer une règle	Ouvre l'Editeur de règles, à partir duquel vous pouvez créer une règle pour le cube sélectionné.
Supprimer la règle	Supprime la règle associée au cube sélectionné. Vous devez détenir les privilèges Admin pour un cube pour supprimer la règle associée.
Exporter comme données ASCII	Exporte les données contenues dans le cube sélectionné dans un fichier ASCII (.cma) à virgules de séparation.
Synchroniser les données	Synchronise les données se trouvant dans le cube sélectionné du serveur de réplication associé.
Sécurité, Réserver	Réserve temporairement le cube sélectionné pour que les autres clients ne puissent pas en modifier les données. Vous devez avoir le privilège Réserve pour pouvoir réserver un cube.
Sécurité, Libérer	Libère le cube que vous avez réservé pour que les autres clients puissent en modifier les données. Vous devez avoir le privilège Réserve pour libérer un cube.
Sécurité, Verrouiller	Verrouille définitivement le cube sélectionné pour que les autres clients ne puissent pas en modifier les données. L'accès à ces éléments sera aussi bloqué pour le client avec lequel vous vous êtes connecté. Vous devez avoir le privilège Verrouillage pour pouvoir verrouiller un cube.
Sécurité, Déverrouiller	Déverrouille le cube sélectionné pour que les autres clients puissent en modifier les données. Vous devez avoir le privilège Admin pour déverrouiller un cube.

Option	Description
Propriétés	Ouvre la boîte de dialogue Propriétés du cube qui permet de définir des dimensions de temps et de mesure.

### Dimensions

Les options suivantes sont disponibles dans le menu Dimensions quand vous sélectionnez un groupe de dimensions dans l'Explorateur de serveur.

Option	Description
Créer une nouvelle dimension	Ouvre la fenêtre de l'Editeur de dimension qui permet de créer une nouvelle dimension.
Editer les attributs	Ouvre la fenêtre de l'Editeur d'attributs qui permet d'affecter et modifier les attributs de toutes les dimensions du groupe sélectionné.
Affectations de sécurité	Ouvre la boîte de dialogue Affectations de sécurité de TM1, depuis laquelle vous pouvez affecter des privilèges de sécurité pour chaque dimension du groupe. Vous devez appartenir au groupe Admin pour utiliser cette option.

### Dimension

Les options suivantes sont disponibles dans le menu Dimension quand vous sélectionnez une dimension dans l'Explorateur de serveur.

Option	Description
Insérer un nouveau sous-ensemble	Ouvre la fenêtre de l'Editeur de sous-ensemble pour la dimension.
Choisir	Copie le nom de la dimension dans le Presse-papiers.
Editer la structure de dimension	Ouvre la dimension sélectionnée pour modification dans la fenêtre de l'Editeur de dimension. Vous devez avoir le privilège Ecriture pour la dimension sélectionnée pour pouvoir utiliser cette option.
Créer une nouvelle dimension	Ouvre une fenêtre vide de l'Editeur de dimension qui permet de créer une nouvelle dimension. Vous devez faire partie du groupe Admin pour pouvoir créer une nouvelle dimension
Exporter une dimension	Exporte les dimensions sélectionnées sous forme de fichier à virgules de séparation (.cma).
Supprimer la dimension	Supprime la dimension sélectionnée. Vous devez appartenir au groupe Admin pour supprimer une dimension

Option	Description
Définir l'ordre des éléments	Ouvre la boîte de dialogue Ordre des éléments de dimension dans laquelle vous pouvez définir l'ordre des éléments de la dimension sélectionnée.
Editer les attributs d'élément	Ouvre la fenêtre de l'Editeur d'attributs qui permet d'affecter et modifier les attributs de tous les éléments de la dimension sélectionnée.
Synchroniser les données	Synchronise les données de la dimension sélectionnée avec les données associées de tous les serveurs répliqués.
Sécurité, Réserver	Réserve temporairement la dimension sélectionnée pour que les autres clients ne puissent pas en modifier la structure. Vous devez avoir le privilège Réserve pour réserver une dimension. Notez que cette option ne réserve que la structure de la dimension. Elle ne réserve aucune des données identifiées par les éléments de cette dimension.
Sécurité, Libérer	Libère une dimension réservée pour que les autres clients puissent en modifier la structure. Vous devez avoir le privilège Réserve pour libérer une dimension. Notez que cette option ne libère que la structure de la dimension. Elle ne libère aucune des données identifiées par les éléments de cette dimension.
Sécurité, Verrouiller	Verrouille en permanence la dimension sélectionnée pour que les autres clients ne puissent pas en modifier la structure. Vous devez avoir le privilège Verrouillage pour verrouiller une dimension. Notez que cette option ne verrouille que la structure de la dimension. Elle ne libère verrouille des données identifiées par les éléments de cette dimension.
Sécurité, Déverrouiller	Déverrouille la dimension sélectionnée pour que les autres clients puissent en modifier la structure. Vous devez avoir le privilège Admin pour pouvoir déverrouiller un élément. Notez que cette option ne verrouille que la structure de la dimension. Elle ne déverrouille des données identifiées par les éléments de cette dimension.
Sécurité, Affectations de sécurité des éléments	Ouvre la boîte de dialogue Affectations de sécurité de TM1, depuis laquelle vous pouvez affecter des privilèges de sécurité pour chaque élément de la dimension. Vous devez avoir le privilège Ecriture pour la dimension sélectionnée pour pouvoir utiliser cette option.

### Vues de cube

Les options suivantes sont disponibles dans le menu Vues de cube quand vous sélectionnez un groupe de vues dans l'Explorateur de serveur.

Option	Description
Créer une nouvelle vue	Ouvre la fenêtre du Cube Viewer qui permet de configurer une nouvelle vue.

### Vue de cube

Les options suivantes sont disponibles dans le menu Vue de cube quand vous sélectionnez une vue dans l'Explorateur de serveur.

Option	Description
Parcourir	Ouvre la vue dans la fenêtre du Cube Viewer.
Parcourir dans Excel	Ouvre la vue dans le Navigateur de feuille de calcul.
Exporter comme données texte	Ouvre la fenêtre Extraction de vue qui vous permet d'exporter la vue comme fichier à virgules de séparation (.cma).
Publier	Cette option est disponible quand vous sélectionnez une vue privée. Choisissez cette option pour convertir une vue privée en vue publique. Les vues publiques sont disponibles pour tous les clients ayant le privilège de lecture pour le cube renfermant la vue.
Supprimer la vue	Supprime la vue sélectionnée. Notez que cette option ne supprime que la définition de vue mais pas les données contenues dans la vue.

### Sous-ensembles

Les options suivantes sont disponibles dans le menu Sous-ensembles quand vous sélectionnez un groupe de sous-ensembles dans l'Explorateur de serveur.

Option	Description
Insérer un nouveau sous-ensemble	Ouvre la fenêtre de l'Editeur de sous-ensemble qui permet de définir un nouveau sous-ensemble.

### Sous-ensemble

Les options suivantes sont disponibles dans le menu Sous-ensemble quand vous sélectionnez un sous-ensemble dans l'Explorateur de serveur.

Option	Description
Ouvrir	Ouvre le sous-ensemble sélectionné dans la fenêtre Editeur de sous-ensemble.



Option	Description
Créer un nouveau sous-ensemble	Ouvre la fenêtre Editeur de sous-ensemble pour la dimension à laquelle le sous-ensemble sélectionné appartient. Cette fenêtre permet de définir de nouveaux sous-ensembles.
Publier	Cette option est disponible quand vous sélectionnez un sous-ensemble privé. Choisissez cette option pour convertir un sous-ensemble de privé à public. Les sous-ensembles publics sont disponibles pour tous les clients ayant le privilège Lecture pour la dimension contenant le sous-ensemble.
Supprimer le sous-ensemble	Supprime le sous-ensemble sélectionné. Notez que cette option ne supprime que la définition de sous-ensemble, mais ne supprime pas les éléments contenus dans le sous-ensemble venant de la dimension parent.

### Répliquations

Les options suivantes sont disponibles dans le menu Répliquations quand vous sélectionnez un groupe de répliquations dans l'Explorateur de serveur.

Option	Description
Insérer une nouvelle répllication	Ouvre la boîte de dialogue Créer un objet de répllication de serveur à partir de laquelle vous pouvez établir une nouvelle connexion de répllication.

### Répllication

Les options suivantes sont disponibles dans le menu Répllication quand vous sélectionnez une répllication dans l'Explorateur de serveur.

Option	Description
Synchroniser les données	Synchronise les données entre les serveurs cible et source.
Modifier les paramètres de répllication	Ouvre la boîte de dialogue Créer un objet de répllication de serveur qui permet de modifier les paramètres de la connexion de répllication sélectionnée.
Supprimer la répllication	Supprime la connexion de répllication sélectionnée.
Afficher les tâches concernées	Ouvre la boîte de dialogue Sélectionner les tâches à modifier. Cette boîte de dialogue permet de supprimer la répllication sélectionnée de toute tâche associée.

### Répliquer un cube

Les options suivantes sont disponibles dans le menu Cube quand vous sélectionnez un cube répliqué dans l'Explorateur de serveur.

Option	Description
Répliquer	Ouvre la boîte de dialogue Répliquer un cube pour le cube sélectionné. Celle-ci permet de définir les paramètres de réplication et de répliquer le cube.
Synchroniser les données	Synchronise les données entre le cube répliqué et le serveur source.

### Processus

Les options suivantes sont disponibles dans le menu Processus quand vous sélectionnez un groupe de processus dans l'Explorateur de serveur.

Option	Description
Créer un nouveau processus	Ouvre TurboIntegrator, dans lequel vous pouvez créer un nouveau processus.
Affectations de sécurité	Ouvre la boîte de dialogue Affectations de sécurité de TM1, depuis laquelle vous pouvez définir les privilèges de sécurité des processus sur le serveur actuel.

### Processus

Les options suivantes sont disponibles dans le menu Processus quand vous sélectionnez un processus dans l'Explorateur de serveur.

Option	Description
Afficher les tâches concernées	Ouvre la boîte de dialogue Sélectionner les tâches à modifier. Cette boîte de dialogue permet de supprimer le processus sélectionné de toute tâche associée.
Modifier un processus	Ouvre le processus sélectionné dans une fenêtre TurboIntegrator.
Exécuter le processus	Exécute le processus sélectionné.
Vue	Affiche un processus en mode de lecture seule. Autorise les membres des groupes DataAdmin et SecurityAdmin à afficher un processus en lecture seule lorsque l'option Accès de sécurité est activée pour le processus.
Accès de sécurité	Contrôle si un processus est autorisé à modifier les données de sécurité dans le script du processus. Seuls les membres de groupes ADMIN et SecurityAdmin sont autorisés à définir cette option. Vous définissez cette option en fonction de chaque processus.

Option	Description
Supprimer un processus	Supprime le processus sélectionné.
Utiliser le bac à sable actif	Configure le processus afin qu'il utilise les données du bac à sable en cours au lieu des données de la base lors de l'exécution du processus. Le bac à sable actif est celui qui est actuellement sélectionné dans le Cube Viewer.

### Tâches

Les options suivantes sont disponibles dans le menu Tâches quand vous sélectionnez un groupe de tâches dans l'Explorateur de serveur.

Option	Description
Créer une nouvelle tâche	Ouvre l'Assistant Définition de tâche à partir duquel vous pouvez planifier une nouvelle tâche.
Affectations de sécurité	Ouvre la boîte de dialogue Affectations de sécurité de TM1, depuis laquelle vous pouvez définir les privilèges de sécurité des tâches sur le serveur actuel.

### Tâche

Les options suivantes sont disponibles dans le menu Tâche quand vous sélectionnez une tâche précise dans l'Explorateur de serveur.

Option	Description
Activer le planning	Cette option bascule le statut d'exécution des tâches. Sélectionnez cette option pour activer l'exécution de la tâche sélectionnée. Une coche apparaît à côté de cette option quand une tâche est activée.  Sélectionnez de nouveau cette option pour désactiver la tâche sélectionnée.
Editer	Ouvre la tâche pour édition dans l'Assistant Définition de tâche.  Vous devez désactiver une tâche avant de la modifier.
Exécuter	Exécute la tâche sélectionnée.
Supprimer	Supprime la tâche sélectionnée.  Vous devez désactiver une tâche avant de la supprimer.

### Menu Edition

Les options ci-après sont disponibles dans le menu Edition de l'Explorateur de serveur.

Option	Description
Copier	Copie l'objet sélectionné dans le Presse-papiers.
Supprimer	Supprime l'objet sélectionné du serveur.

## Menu Affichage

Les options ci-après sont disponibles dans le menu Affichage de l'Explorateur de serveur.

Option	Description
Barre de statut	Masque ou affiche la barre de statut au bas de la fenêtre de l'Explorateur de serveur. Une coche indique que la barre de statut est affichée.
Barre d'outils	Masque ou affiche la Barre d'outils en haut de la fenêtre de l'Explorateur de serveur. Une coche indique que la barre d'outils est affichée.
Fenêtre Propriétés	Masque ou affiche la sous-fenêtre Propriétés de l'Explorateur de serveur. Une coche indique que la sous-fenêtre Propriétés est affichée.
Objets : Applications Cubes Dimensions Répliquions Processus Tâches	Masque ou affiche l'un des objets de la sous-fenêtre de gauche de l'Explorateur de serveur (Arborescence).  Une coche indique que l'objet sélectionné est affiché.
Réduire tous les enfants	Contracte l'arborescence dans la sous-fenêtre gauche de l'Explorateur de serveur pour cacher tous les enfants d'un objet sélectionné.
Développer tous les enfants	Développe l'arborescence dans la sous-fenêtre gauche de l'Explorateur de serveur pour montrer tous les enfants d'un objet sélectionné.
Afficher les objets de contrôle	Masque ou affiche les cubes de contrôle dans la sous-fenêtre gauche de l'Explorateur de serveur. Une coche indique que les objets de contrôle sont affichés.
Actualiser	Met à jour l'affichage hiérarchique en cours des objets dans la sous-fenêtre gauche de l'Explorateur de serveur.

## Editeur de sous-ensemble

### Sous-fenêtre Eléments

Affiche une représentation hiérarchique de tous les éléments du sous-ensemble que vous visualisez actuellement.

### Sous-fenêtre Propriétés

Affiche les propriétés de l'élément sélectionné dans la sous-fenêtre Eléments de l'Editeur de sous-ensemble. Quand vous sélectionnez un élément consolidé, cette sous-fenêtre affiche le nom, le type et la pondération de tous les enfants de l'élément consolidé.

**Remarque :** Lorsque vous affichez un ensemble de dimensions de taille exceptionnelle dans l'Editeur de sous-ensemble et que la sous-fenêtre de Propriétés est active, il se peut que vous rencontriez des problèmes de performance. Cela peut se produire lorsque vous sélectionnez une consolidation dans la sous-fenêtre des éléments et que TM1 doit afficher la liste complète des propriétés et éléments apparentés dans la sous-fenêtre Propriétés.

Si vous travaillez avec des ensembles de dimensions de grande taille, vous voudrez peut-être désactiver la sous-fenêtre Propriétés. Pour désactiver la sous-fenêtre Propriétés, cliquez sur l'option Fenêtre de propriétés dans le menu Affichage pour supprimer la coche qui se trouve à côté de l'option.

## Menu Sous-ensemble

Élément de menu	Description
Ouvrir	Ouvre la boîte de dialogue Enregistrer un sous-ensemble de TM1. Sélectionnez un sous-ensemble dans la liste et cliquez sur OK pour l'ouvrir.
Recharger	Recharge la définition de sous-ensemble en cours.
Enregistrer	Enregistre la définition de sous-ensemble en cours.
Enregistrer sous	Enregistre la définition de sous-ensemble en cours sous un nouveau nom.
Fermer	Ferme l'Editeur de sous-ensemble.

## Menu Edition

Élément de menu	Description
Annuler	Annule la dernière action.
Rétablir	Rétablit la dernière action "Annuler".
Couper	Coupe les éléments sélectionnés et les met dans le Presse-papiers.
Copier	Copie les éléments sélectionnés dans le Presse-papiers.

<b>Élément de menu</b>	<b>Description</b>
Copier le nom unique	Copie le nom de l'élément dans le Presse-papiers, sous forme d'expression MDX. Le nom copié peut alors être collé dans la fenêtre d'expression de l'Editeur de sous ensemble.
Coller	Colle le contenu du Presse-papiers au point d'insertion en cours.
Coller au-dessus	Colle le contenu du Presse-papiers au-dessus de l'élément sélectionné actuellement.
Coller en dessous	Colle le contenu du Presse-papiers en dessous de l'élément sélectionné actuellement.
Insérer un sous-ensemble	Ouvre une nouvelle instance de l'Editeur de sous-ensemble pour vous permettre d'ajouter une consolidation utilisateur au sous-ensemble en cours.
Garder	Ne garde que les éléments actuellement sélectionnés dans la sous-fenêtre Eléments de l'Editeur de sous-ensemble et enlève tous les autres.
Supprimer	Supprime les éléments sélectionnés de la définition de sous-ensemble en cours.
Choisir les éléments, Horizontal	Copie les éléments sélectionnés dans le Presse-papiers avec une orientation horizontale, ce qui permet de les coller sur une ligne de feuille de calcul.
Choisir les éléments, Vertical	Copie les éléments sélectionnés dans le Presse-papiers avec une orientation verticale, ce qui permet de les coller dans une colonne de feuille de calcul.
Trier par ordre décroissant	Trie tous les éléments de la sous-fenêtre Eléments par ordre alphabétique décroissant.
Trier par ordre croissant	Trie tous les éléments de la sous-fenêtre Eléments par ordre alphabétique croissant.
Trier par hiérarchie	Trie tous les éléments dans la sous-fenêtre Eléments par ordre hiérarchique, pour afficher les relations parent/enfant entre les éléments.
Trier par index croissant	Trie tous les éléments de la sous-fenêtre Eléments par ordre croissant selon la valeur d'index d'élément.

<b>Élément de menu</b>	<b>Description</b>
Trier par index décroissant	Trie tous les éléments de la sous-fenêtre Eléments par ordre décroissant selon la valeur d'index d'élément.
Passer au niveau inférieur	Affiche les enfants immédiats des éléments sélectionnés.
Regrouper	Affiche les parents immédiats des éléments sélectionnés.
Développer un élément	Affiche tous les enfants des éléments sélectionnés.
Réduire un élément	Réduit l'affichage des consolidations sélectionnées de telle sorte que les enfants ne sont pas affichés.
Filtrer par niveau	Ouvre la boîte de dialogue Filtrer par niveau qui permet de sélectionner des éléments par niveau de hiérarchie.
Filtrer par attribut	Ouvre la boîte de dialogue Filtrer par attribut qui permet de sélectionner des éléments par valeur d'attribut.
Filtrer par extraction de vue	Permet de sélectionner uniquement les éléments qui correspondent à une requête définie par l'utilisateur.  Cette option n'est disponible que quand vous ouvrez l'Editeur de sous-ensemble en cliquant sur une étiquette de dimension dans la fenêtre Cube Viewer.
Filtrer par caractère générique	Permet de sélectionner des éléments correspondant à une chaîne de recherche définie par l'utilisateur.
Sélectionner un alias	Ouvre la boîte de dialogue Alias de TM1, qui vous permet de sélectionner un alias défini précédemment en fonction du quel les noms d'élément doivent être affichés.
Sécurité, Réserver	Réserve temporairement l'élément sélectionné pour que les autres clients ne puissent pas modifier les données identifiées par cet élément. Vous devez avoir le privilège Réserve pour pouvoir réserver un élément.
Sécurité, Libérer	Libère un élément sélectionné pour que les autres clients puissent modifier les données identifiées par cet élément. Vous devez avoir le privilège Réserve pour pouvoir libérer un élément.

<b>Élément de menu</b>	<b>Description</b>
Sécurité, Verrouiller	Verrouille définitivement l'élément sélectionné pour que les autres clients ne puissent pas modifier les données identifiées par cet élément. Vous devez avoir le privilège Verrouillage pour pouvoir verrouiller un élément.
Sécurité, Déverrouiller	Déverrouille l'élément sélectionné pour que les autres clients puissent modifier les données identifiées par cet élément. Vous devez avoir le privilège Admin pour pouvoir déverrouiller un élément.
Editer des formats d'éléments	Ouvre la feuille de calcul Editer les formats d'éléments, dans laquelle vous pouvez définir des styles d'affichage pour des tranches dynamiques et des feuilles Web TM1.

## Menu Affichage

<b>Élément de menu</b>	<b>Description</b>
Barres d'outils	Ouvre un sous-menu qui permet d'activer ou de désactiver l'affichage de toutes les barres d'outils de l'Editeur de sous-ensemble.  Une coche indique qu'une barre d'outils est affichée.
Barre de statut	Masque ou affiche la barre de statut au bas de la fenêtre de l'Editeur de sous-ensemble.  Une coche indique que la barre de statut est affichée.
Fenêtre Propriétés	Affiche ou masque la sous-fenêtre Propriétés.  Une coche indique que la sous-fenêtre Propriétés est affichée.
Fenêtre d'expression	Masque ou affiche la Fenêtre d'expression au bas de l'Editeur de sous-ensemble. Une coche indique que la Fenêtre d'expression est affichée.



<b>Élément de menu</b>	<b>Description</b>
Développer au-dessus	<p>Cette option détermine comment les consolidations se développent et se contractent lorsque vous passez au niveau inférieur.</p> <p>Quand cette option est activée, les enfants d'une consolidation sont développés au dessus de la consolidation quand vous passez au niveau inférieur.</p> <p>Quand cette option est désactivée, les enfants d'une consolidation sont développés en dessous de la consolidation quand vous passez au niveau inférieur.</p> <p>Quand l'option Développer au-dessus est activée dans un sous-ensemble, le passage au niveau inférieur d'une consolidation dans Cube Viewer, le navigateur de feuille de calcul ou une tranche produit le comportement suivant :</p> <p>Quand cette option est activée dans un sous-ensemble en ligne, le passage au niveau inférieur d'une consolidation affiche les enfants au-dessus de la consolidation.</p> <p>Si cette option est activée dans un sous-ensemble en colonne, le passage au niveau inférieur d'une consolidation affiche les enfants à gauche de cette consolidation.</p>
Actualiser	Met à jour l'affichage de la sous-fenêtre Eléments.

## Menu Outils

<b>Élément de menu</b>	<b>Description</b>
Enregistrer une expression	Lance l'enregistrement de vos actions dans l'Editeur de sous-ensemble.
Arrêter l'enregistrement	<p>Arrête l'enregistrement de vos actions dans l'Editeur de sous-ensemble.</p> <p>Lorsque vous arrêtez l'enregistrement, TM1 génère une expression MDX qui peut être enregistrée pour créer un sous-ensemble dynamique.</p>
Effacer une expression	Efface le contenu de la Fenêtre d'expression.
Filtre	Ouvre la boîte de dialogue Filtrer un sous-ensemble qui permet de créer un sous-ensemble dynamique basé sur des valeurs de cube.

## Boîte de dialogue Alias

Pour afficher les éléments du sous-ensemble en cours d'après les alias affectés, sélectionnez un nom d'alias dans la liste et cliquez sur **OK**.

## Boîte de dialogue Options TM1

Les options suivantes peuvent être définies dans la boîte de dialogue Options TM1.

### Paramètres de connexion

Option	Description
Hôte admin	Entrez le nom d'ordinateur de votre hôte Admin. L'hôte Admin est l'ordinateur sur lequel s'exécute votre serveur Admin.
Connexion intégrée	Basculez cette option pour l'activer et utiliser la connexion intégrée. Désactivez cette option pour utiliser la sécurité des connexions TM1 standard. Par défaut, l'option est désactivée.

### Serveur local

Option	Description
Répertoire de données du serveur local	Entrez le chemin complet de votre Répertoire de données du serveur local, ou cliquez sur le bouton Parcourir adjacent pour accéder au répertoire. Vous pouvez aussi cliquer sur la flèche vers le bas pour faire une sélection dans la liste des répertoires ouverts récemment.
Se connecter au serveur local au démarrage	Désactivez cette option pour lancer TM1 Perspectives/TM1 Architect sans lancer le serveur local. La valeur par défaut est option activée.

**Remarque :** Le serveur local est pris en charge uniquement sur les versions 32 bits de TM1. Le répertoire de données par défaut pour le serveur local est Pdata. Si vous utilisez une version 64 bits de TM1, le serveur exemple Sdata installé par défaut avec le serveur TM1, contient les mêmes objets et données que Pdata.

### Protocole TSL (Transport Layer Security) du serveur Admin

<b>Option</b>	<b>Description</b>
Autorité de certification	Chemin complet du fichier d'autorité de certification qui a délivré le certificat du serveur Admin.
Liste de révocation de certificats	Chemin complet du fichier de révocation de certificat émis par l'autorité de certification ayant délivré initialement le certificat du serveur Admin. Ce fichier n'existe que si un certificat a été révoqué.
ID de certificat	Nom du principal à qui le certificat du serveur Admin est délivré.
Utiliser le magasin de certificats	Sélectionnez cette option pour que le certificat de l'autorité de certification qui a délivré initialement le certificat du serveur Admin soit exporté depuis le magasin de certificats Windows lors de l'exécution.  Quand cette option est sélectionnée, vous devez aussi définir une valeur pour Exporter l'ID du certificat dans la boîte de dialogue Options de TM1.
ID du certificat d'exportation	Clé d'identité utilisée pour exporter le certificat de l'autorité de certification qui a initialement délivré le certificat du serveur Admin et provenant du magasin de certificats.  Requis uniquement si vous activez l'option Utiliser le magasin de certificats.

## Boîte de dialogue Requête du journal des transactions

La boîte de dialogue Requête du journal des transactions permet d'interroger et de visualiser les enregistrements figurant dans le journal des transactions de TM1 (Tm1s.log). Cette boîte de dialogue contient les zones de quatre paramètres que vous devez spécifier pour exécuter une requête.

<b>Option</b>	<b>Description</b>
Heure de début	La date et l'heure de début de la requête.  TM1 interroge tous les enregistrements consignés au journal des transactions à partir de ces date et heure. Vous devez utiliser le format MM/JJ/AAAA HH:MM:SS pour spécifier une heure de début.  Par défaut, l'horodatage de début est 00:01:00 GMT pour la date à laquelle la requête est lancée.

Option	Description
Heure de fin	La date/heure de fin de la requête. La valeur par défaut, __/__/______:__:__, représente une date et une heure indéfinies. Si vous acceptez cette valeur par défaut, TM1 interroge tous les enregistrements jusqu'à la date et l'heure de lancement de la requête.
Client(s)	Le ou les clients pour lesquels la requête est exécutée. Vous pouvez exécuter cette requête pour un client unique ou pour tous les clients. Le réglage par défaut est tous les clients (*).
Cube(s)	Le ou les cubes pour lesquels la requête est exécutée. Vous pouvez exécuter cette requête pour un cube unique ou pour tous les cubes. Le réglage par défaut est tous les cubes (*).

Pour définir un des paramètres décrits ci-dessus, cliquez sur la flèche en face de la zone appropriée.

## Boîte de dialogue Résultats de la requête du journal des transactions

La boîte de dialogue Résultats de la requête du journal des transactions affiche les résultats de requête du journal des transactions sous forme de table. Cette table contient les colonnes suivantes pour chaque enregistrement renvoyé par la requête :

En colonnes	Description
LOGTIME	L'heure à laquelle une valeur a été modifiée.
REPLICATIONTIME	L'heure à laquelle une valeur a été répliquée.
CLIENT	Le nom du client qui a écrit la valeur.
OLDVALUE	La valeur des données avant la modification.
NEWVALUE	La valeur des données après la modification.
CUBENAME	Le cube dans lequel la valeur a été écrite.
KEY N	Il y a plusieurs colonnes Key N dans la table, chaque colonne représentant les éléments qui identifient la valeur.

La boîte de dialogue Résultats de requête du journal des transactions comprend trois menus.

Le menu **Fichier** contient une seule commande : Quitter.

Le menu **Aide** contient une commande qui permet d'ouvrir l'aide pour la boîte de dialogue.

Le menu **Edition** contient les commandes suivantes :

Élément de menu	Description
Copier	Copie une cellule sélectionnée unique dans le Presse-papiers
Masquer	Supprime l'affichage des enregistrements sélectionnés dans la table. Cliquez sur Rafraîchir pour rétablir l'affichage des enregistrements cachés.
Trier	Ouvre un sous-menu dans lequel vous pouvez choisir les colonnes à trier ou un ordre de tri à appliquer.
Rechercher	Ouvre la boîte de dialogue Rechercher/Remplacer qui permet de faire des recherches dans la table en cours.
Sélectionner	Sélectionne les enregistrements en surbrillance
Désélectionner	Désélectionne les enregistrements en surbrillance.
Sélectionner tout	Sélectionne tous les enregistrements de la table.
Désélectionner tout	Désélectionne tous les enregistrements de la table.
Restaurer	Revient aux versions antérieures des enregistrements sélectionnés. Quand vous revenez à la version antérieure d'un enregistrement, sa valeur OLDVALUE remplace la valeur NEWVALUE.  Quand plusieurs enregistrements sont sélectionnés pour un emplacement de cube unique, les valeurs OLDVALUE sont prises dans les valeurs LOGTIME les plus anciennes.

## Editeur TurboIntegrator

L'Editeur TurboIntegrator permet de définir des processus permettant d'importer des données ou des métadonnées depuis plusieurs sources possibles. L'Editeur comprend cinq onglets, dont plusieurs sont dynamiques ou contiennent des sous-onglets. Vous définissez un processus en remplissant chaque onglet dans un ordre donné.

### Menu Fichier

Élément de menu	Description
Enregistrer	Enregistre la définition de processus en cours.
Enregistrer sous	Enregistre la définition de processus en cours sous un nouveau nom.

Elément de menu	Description
Exécuter	Exécute le processus en cours.
Quitter	Ferme l'Editeur TurboIntegrator.

## Menu Edition

Elément de menu	Description
Annuler	Annule la dernière action de frappe effectuée dans le sous-onglet de procédure Prologue, Métadonnées, Données ou Epilogue.
Couper	Coupe le texte sélectionné et le met dans le Presse-papiers.
Copier	Copie le texte sélectionné et le met dans le Presse-papiers.
Coller	Colle le contenu du Presse-papiers dans la zone ou la cellule en cours.

## Onglet Source de données

Servez-vous de l'onglet Source de données pour identifier et accéder à la source dont vous voulez importer des données.

**Remarque :** Pendant la définition d'un processus à partir du client TM1, le chemin d'accès à une source de données ASCII ou ODBC peut ne pas correspondre au chemin utilisé par le serveur. Dans ce cas, le processus échoue. Pour assurer le fonctionnement correct du processus :

- Définissez les processus impliquant des sources de données ODBC sur le serveur effectif sur lequel résidera le processus. Veillez à ne pas utiliser de client distant pour définir un tel processus.
- Servez-vous du Voisinage réseau de Windows pour définir le chemin aux sources de données ASCII. Ainsi, il n'y a pas de confusion possible dans le chemin d'accès aux clients et aux serveurs.

Les zones et les options disponibles sur l'onglet Source de données varient selon le Type de source de données que vous sélectionnez. Les tableaux ci-dessous décrivent les zones requises et les options de chaque source.

### ODBC

Définissez une source de données ODBC :

Zones	Description
Nom de la source de données	Le chemin d'accès complet de la source de données ODBC.
Nom d'utilisateur	Votre nom d'utilisateur sur la source.
Mot de passe	Votre mot de passe.
Requête	Une requête SQL qui servira à extraire les données de la source.

Zones	Description
Utiliser Unicode	Cochez cette case si vous souhaitez utiliser le format Unicode pour cette source.
Aperçu	Affiche les dix premiers enregistrements.

### Texte

Définissez une source de données ASCII ou Texte :

Zones	Description
Nom de la source de données	Chemin d'accès complet du fichier texte source. Pour être certain que ce chemin d'accès sera reconnu par le client et le serveur, cliquez sur le bouton Parcourir et servez-vous du Voisinage réseau pour définir le chemin d'accès.
Nom de la source de données sur le serveur	Lorsque vous créez un nouveau processus, TurboIntegrator suppose que le nom de la source de données sur le serveur est identique au nom de la source de données utilisée pour créer le processus.  Si le nom de la source de données sur le serveur est différent de celui de la source de données locale utilisée pour créer le processus entrez le chemin d'accès absolu du fichier de source de données sur le serveur.
Type de limiteur	Si la source utilise un caractère pour définir les colonnes, sélectionnez <b>Délimité</b> puis choisissez le caractère approprié dans la zone <b>Délimiteur</b> .
Largeur fixe	Si la source utilise une largeur fixe, sélectionnez <b>Largeur fixe</b> puis, à l'aide du bouton <b>Définir la largeur de zone</b> , ouvrez la boîte de dialogue Aperçu afin de définir les largeurs de colonnes.
Type de guillemets	Spécifiez le type de guillemet utilisé dans la source de données.
Nombre d'enregistrements de titre	Si les enregistrements de titre s'étendent sur plusieurs lignes, indiquez ici le nombre de lignes. Sinon, laissez cette zone vide.
Délimiteurs de nombres	Entrez le caractère à utiliser comme <b>Séparateur décimal</b> et comme <b>Séparateur de milliers</b> dans la source.

### ODBO

Définissez une source de données ODBC établie pour une base de données relationnelles qui peut être créée à l'aide du panneau de configuration Sources de données de Microsoft Windows :

### Cube ou dimension

Sélectionnez le type d'importation à effectuer :

<b>Zones</b>	<b>Description</b>
Fournisseur ODBO	Pour importer un cube ODBO, sélectionnez la source de données OLE DB pour OLAP à utiliser dans la liste déroulante, ainsi que Microsoft Analysis Services.
Emplacement ODBO	Indiquez l'emplacement du fichier de ce fournisseur. Si vous importez une dimension, laissez cette zone vide.
Source de données ODBO	Entrez le nom affecté par l'administrateur à un ensemble de catalogues à un emplacement particulier. Dans Microsoft Analysis Services, il s'agit du nom d'un serveur enregistré.
Catalogue ODBO	Nom affecté par votre administrateur à une collection particulière de bases de données (cubes, dimensions et autres objets). Pour MAS, il s'agit du nom de la base de données.
ID utilisateur ODBO	Nom d'utilisateur valide pour la base de données.
Mot de passe ODBO	Mot de passe valide dans cette source de données.
Paramètres de connexion supplémentaires	Entrez ici tout autre paramètre requis par la source ODBO. Séparez ces paramètres par des points-virgules. Si vous importez une dimension, n'entrez aucune valeur.

### **Requête MDX**

Définissez une requête MDX :

<b>Zones</b>	<b>Description</b>
Onglet Connexion	Entrez les informations de connexion demandées. Les paramètres de connexion sont spécifiques au fournisseur. Cliquez sur Connecter. Si la connexion réussit, l'onglet Requête MDX est activé.
Onglet Requête MDX	Entrez la requête MDX dans cet onglet ou collez-en une provenant d'une autre application.

### **SAP**

Définit la source de données SAP RFC :



Onglet	Zone	Description
Connexion	Système	Le nom du système SAP auquel vous souhaitez vous connecter. Si le nom du système comporte des espaces, vous devez le placer entre des guillemets doubles.
	Client	Un nombre correspondant à la version de l'interface utilisateur sur le serveur SAP. Par exemple, 498.
	Utilisateur	Votre nom d'utilisateur sur le système SAP.
	Mot de passe	Votre mot de passe sur le système SAP.
	Langue	La langue que vous souhaitez utiliser pour vous connecter au système SAP. Toutes les descriptions textuelles sont renvoyées dans la langue spécifiée, si elle est disponible. Le paramètre de langue est une abréviation de deux lettres, par exemple, EN=Anglais.
	Paramètres de connexion supplémentaires	Entrez, le cas échéant, les autres paramètres et valeurs que vous utilisez pour vous connecter à votre système SAP BW.
	Taille du paquet	Valeur limitant le nombre de lignes dans chaque paquet envoyé depuis SAP vers TM1. Les paquets de plus petite taille entraînent un trafic accru de petits paquets sur le réseau, tandis que les paquets de plus grande taille entraînent une réduction du trafic, mais avec de plus grand paquets dans chaque transmission. La taille du paquet par défaut, qui est aussi la taille de paquet minimum, est de 50 000.

## InfoCube

Aire	Zone	Description
InfoCube	Afficher les noms techniques SAP	Pour employer les noms techniques, cochez cette case. Ne la cochez pas si vous souhaitez afficher les noms descriptifs.
	Sélectionner un InfoCube à charger depuis	Cette option permet d'indiquer l'InfoCube à partir duquel importer les données.
	Sélectionnez le cube TM1 vers lequel effectuer le chargement	Pour importer l'InfoCube SAP dans un cube TM1 existant, cliquez sur cette option et sélectionnez le cube devant recevoir les données de l'InfoCube SAP.
	Sélectionnez le cube TM1 vers lequel effectuer le chargement	Pour créer un cube TM1 lors de l'importation de l'InfoCube, indiquez dans cette zone le nom du nouveau cube TM1.
Action de cube TM1	Créer	Importe les données et les métadonnées depuis l'InfoCube SAP et crée un nouveau cube dans TM1. Utilisez cette option uniquement lorsque les cubes et les dimensions importés n'existent pas sur le serveur.
	Recréer	Détruit un cube TM1 existant et le régénère à partir des données et des métadonnées de l'InfoCube SAP. Servez-vous de cette option uniquement lorsque le cube et les dimensions TM1 existent et que vous souhaitez les remplacer par de nouvelles structures et de nouvelles données provenant de l'InfoCube SAP.
	Mettre à jour	Importe les données depuis un InfoCube SAP et les insère dans un cube TM1 existant. Cette option ne modifie pas la structure des cubes et des dimensions sur le serveur.

Aire	Zone	Description
		<p>Processus spécifiant que l'option Pas d'action n'affecte pas les données ou les métadonnées des cubes TM1.</p> <p>Utilisez cette option pour tester et déboguer des processus ou pour définir vos propres opérations.</p>
Action de données	Stocker les valeurs	<p>Cette option écrit des valeurs de cellule provenant de l'InfoCube SAP dans le cube TM1.</p> <p>Si vous sélectionnez cette option lorsque l'option Mettre à jour un cube est sélectionnée, les valeurs de cube TM1 existantes sont écrasées par les valeurs importées depuis l'InfoCube.</p>
	Accumuler les valeurs	<p>L'option Accumuler les valeurs vous permet d'agréger les valeurs existant dans le cube TM1 avec celles importées de l'InfoCube SAP.</p>
	Remplir de zéro la portion du cube cible	<p>Cette option devient disponible quand vous sélectionnez l'action Mettre un cube à jour.</p> <p>Sélectionnez cette option si vous voulez définir toutes les données individuelles d'une vue à zéro.</p> <p>Pour définir la vue de cube à zéro, vous pouvez faire ce qui suit :</p> <ul style="list-style-type: none"> <li>• Cliquez sur la liste Vue pour sélectionner une vue existante à mettre à zéro.</li> <li>• Cliquez sur <b>Plus</b> à côté de l'option Vue pour définir une nouvelle vue à mettre à zéro.</li> </ul>

Aire	Zone	Description
	Activer le journal de cube	<p>Pour consigner les modifications des données de cube lors de l'importation depuis un InfoCube SAP, sélectionnez cette option.</p> <p>Pour désactiver la journalisation lors de l'importation, désactivez cette option.</p> <p><b>Remarque :</b> La désactivation de la journalisation accélère le chargement et la mise à jour de données mais rend impossible la récupération des mises à jour en cas de panne du système.</p>

### Onglet Caractéristique

Zone	Description	
Sélectionner des hiérarchies	Identifier les hiérarchies dans la source de données.	Identifier les hiérarchies dans la source de données.
Date d'évaluation	<p>Date à laquelle tous les attributs SAP qui dépendent de la date sont importés dans TM1 tels qu'ils étaient à la date spécifiée. Les attributs qui ne dépendent pas de la date sont importés sous la forme sous laquelle ils étaient lors de l'exécution du processus.</p> <p>Si cette date n'est pas spécifiée, <i>tous</i> les attributs SAP sont importés tels qu'ils existent à la date d'exécution du processus TM1.</p> <p>N'importez pas de hiérarchie comptant des intervalles.</p>	<p>Date à laquelle tous les attributs SAP qui dépendent de la date sont importés dans TM1 tels qu'ils étaient à la date spécifiée. Les attributs qui ne dépendent pas de la date sont importés sous la forme sous laquelle ils étaient lors de l'exécution du processus.</p> <p>Si cette date n'est pas spécifiée, <i>tous</i> les attributs SAP sont importés tels qu'ils existent à la date d'exécution du processus TM1.</p> <p>N'importez pas de hiérarchie comptant des intervalles.</p>
Dimension TM1	<p>Sélectionnez la dimension TM1 existante mappée avec cette caractéristique.</p> <p>Laissez cette zone vide si vous ne désirez pas importer la caractéristique dans votre cube TM1.</p>	<p>Sélectionnez la dimension TM1 existante mappée avec cette caractéristique.</p> <p>Laissez cette zone vide si vous ne désirez pas importer la caractéristique dans votre cube TM1.</p>
Action de dimension TM1	Créer	Crée une dimension TM1 depuis la caractéristique SAP.

Zone	Description	
	Recréer	Recrée complètement une dimension TM1 avec des éléments importés depuis la caractéristique SAP.
	Mettre à jour	Met à jour une structure de dimension existante en ajoutant de nouveaux éléments importés depuis la caractéristique SAP.
	Telle quelle	<p>Traite la caractéristique avec TurboIntegrator, mais ne l'utilise pas pour créer ou modifier des dimensions TM1.</p> <p>Servez-vous de cette option pour tester et déboguer les processus ou manipuler la caractéristique dans l'onglet Avancé de TurboIntegrator.</p>
	Ne pas charger	Permet de ne pas importer la caractéristique SAP dans TM1. La caractéristique est exclue entièrement lorsque l'InfoCube SAP est traité via TurboIntegrator.
Sélectionner des attributs	Attributs de caractéristique	Définit les attributs de cette source de données.
	Texte	Identifie les attributs par une valeur de chaîne.
	Numérique	Identifie les attributs par une valeur numérique
	Alias	Identifie les attributs qui sont des noms alternatifs pour les dimensions auxquelles ils sont associés. Les alias de dimension doivent être uniques et n'être identiques à aucun autre alias ou nom de dimension.
Chiffres clés	<p>Sélectionnez chaque chiffre clé à importer dans TM1.</p> <p>Si les chiffres clés sont mappés vers une dimension TM1 existante, cliquez sur la colonne Dimension de <b>TM1</b> et sélectionnez celle correspondant aux chiffres clés.</p>	<p>Sélectionnez chaque chiffre clé à importer dans TM1.</p> <p>Si les chiffres clés sont mappés vers une dimension TM1 existante, cliquez sur la colonne Dimension de <b>TM1</b> et sélectionnez celle correspondant aux chiffres clés.</p>

<b>Zone</b>	<b>Description</b>	
Restrictions	Ajouter la Restriction	Crée une nouvelle restriction pour cette caractéristique.
	Caractéristique SAP	Sélectionnez les caractéristiques sur lesquelles définir une restriction.
	Signe	Indique si la restriction est inclusive ou exclusive. Sélectionnez Inclure si vous souhaitez que le processus TurboIntegrator importe seulement les valeurs répondant à la définition de restriction. Sélectionnez Exclure si vous souhaitez que le processus TurboIntegrator importe seulement les valeurs ne répondant pas à la définition de restriction.
	Option	Opérateur employé pour la restriction. Vous pouvez choisir entre huit opérateurs, comme décrit dans la table ci-dessous.

### Opérateurs de restriction

<b>Opérateur</b>	<b>Description</b>
=	La restriction identifie uniquement les caractéristiques égales à la Valeur inférieure spécifiée.
< >	La restriction identifie uniquement les caractéristiques inférieures ou supérieures à la Valeur inférieure spécifiée.
<	La restriction identifie uniquement les caractéristiques inférieures à la Valeur inférieure spécifiée.
>	La restriction identifie uniquement les caractéristiques supérieures à la Valeur inférieure spécifiée.
< =	La restriction identifie uniquement les caractéristiques inférieures ou égales à la Valeur inférieure spécifiée

Opérateur	Description
> =	La restriction identifie uniquement les caractéristiques supérieures ou égales à la Valeur inférieure spécifiée
[ ]	La restriction identifie uniquement les caractéristiques se trouvant entre la Valeur inférieure et la Valeur supérieure spécifiées (ces deux valeurs étant incluses).
] [	La restriction identifie uniquement les caractéristiques se trouvant hors de la Valeur inférieure et la Valeur supérieure spécifiées (ces deux valeurs étant incluses).

Vous pouvez choisir entre huit opérateurs, comme décrit dans la table ci-dessous.

Entrez une valeur faible pour la restriction dans la colonne Valeur inférieure.

Entrez une valeur haute pour la restriction dans la colonne Valeur supérieure.

**Remarque :** Les restrictions ne sont pas validées via Vous TurboIntegrator. Vous devez vous assurer que les restrictions que vous entrez sont correctes et valides pour vos données SAP.

### Sécurité

Zone	Description
Importation de la sécurité	Indique que les affectations de sécurité correspondant à cette caractéristique doivent être importées.
Consolidation de premier niveau	Crée une consolidation de premier niveau pour la dimension TM1 créée à l'aide du nom saisi ici.

Zone	Description
Rendre les textes uniques	<p>Sélectionnez cette option pour générer des alias uniques pour tous les éléments créés à partir de la caractéristique SAP. Quand vous importez une caractéristique SAP dans TM1, les valeurs de la caractéristique deviennent des éléments de dimension TM1 et les descriptions de valeurs SAP deviennent des alias d'éléments TM1. Dans TM1, tous les alias d'éléments d'une dimension doivent être uniques. Si un processus TurboIntegrator tente d'affecter le même alias à plusieurs éléments, le processus génère des erreurs et la création d'alias échoue.</p> <p>Lorsque l'option Rendre les textes uniques est sélectionnée, TM1 examine les descriptions SAP qui sont importées et converties en alias TM1. Si TM1 détecte que plusieurs valeurs utilisent des descriptions identiques, TM1 ajoute le nom de la valeur à la description pour générer des alias uniques.</p> <p>Si la case <b>Rendre les textes uniques</b> n'est pas cochée, aucun donnée SAP_Text n'est extraite.</p> <p>Lorsque la case <b>Rendre les textes uniques</b> n'est pas cochée, vous pouvez ajouter <code>DataSourceSAPUsingTexts=1;</code> au prologue pour importer les valeurs de l'attribut d'alias.</p> <p>Pour obtenir le même comportement à l'aide d'un script TI, reportez-vous à la section relative à la fonction AttrPutS.</p>
Date d'évaluation	<p>Toutes les valeurs de caractéristique se trouvant entre la date sélectionnée et la date d'exécution du processus seront importées dans TM1. Si aucune date d'évaluation n'est précisée, la date par défaut est la date d'exécution du processus TurboIntegrator.</p>

#### Table SAP

Zone	Description
Table SAP	Indique que la source de données est une requête de table SAP.
Nom de la table	Nom de la table SAP à utiliser.
Chaîne de filtre	Chaîne de filtre SQL à utiliser dans la clause WHERE lorsque l'instruction SQL SELECT générée par TurboIntegrator est exécutée dans la table SAP.



## Table ODS

Zone	Description
Table ODS	Permet d'exporter les données TM1 dans une table ODS pouvant ensuite servir à importer les données via un InfoCube SAP.
Configuration de ODS	Définissez les détails de la table ODS.
Parcourir	Sélectionnez la vue TM1 à utiliser comme source de données.
Afficher les noms techniques	Pour employer les noms techniques, cochez cette case. Ne la cochez pas si vous souhaitez afficher les noms descriptifs.
Sélectionnez une table ODS	Sélectionnez la table ODS vers laquelle exporter les données.
Colonnes	<p>Les colonnes peuvent être des caractéristiques SAP ou des chiffres clés. Vous devez bien connaître la structure de table ODS pour savoir quelles colonnes sont des caractéristiques ODS et quelles colonnes sont des chiffres clés.</p> <p>Prenez note des points suivants lorsque vous mappez des dimensions avec des caractéristiques :</p> <ul style="list-style-type: none"> <li>• Vous n'avez pas à mapper une dimension sur toutes les colonnes de caractéristiques de la table ODS. Certaines colonnes peuvent ne pas avoir de dimension TM1 correspondante une fois le mappage terminé. Dans ce cas, toute colonne de caractéristique qui n'est pas mappée sera vide à l'issue de l'exportation.</li> <li>• Vous ne devez pas mapper une seule dimension TM1 avec plusieurs colonnes de table de caractéristiques ODS. L'interface utilisateur de TurboIntegrator ne vous empêche pas de le faire, mais un tel mappage aboutira à des valeurs de colonnes redondantes dans la table ODS.</li> <li>• Lorsque vous mappez une dimension en titre de vue TM1 vers une caractéristique et que la dimension en titre <i>n'utilise pas</i> un sous-ensemble nommé, seul le dernier élément du sous-ensemble en titre non nommé en cours est exporté vers la table ODS. Si la dimension en titre <i>n'utilise pas</i> un sous-ensemble nommé, tous les éléments de sous-ensemble sont exportés vers la table ODS.</li> </ul>

Zone	Description
Sélectionner une mesure	<p>La dernière dimension de la vue de cube source est supposée être la dimension de mesures. Quand vous mappez une dimension de mesures avec une colonne de table ODS, le bouton Sélectionner une mesure devient disponible.</p> <p>Sélectionnez l'élément unique mappé directement avec la colonne des chiffres clés de la table ODS.</p> <p>Si votre table ODS comprend une seule colonne de chiffres clés unique, vous pouvez aussi utiliser les chiffres clés secondaires.</p>
Dimension TM1	<p>Si votre table ODS comprend une seule colonne de chiffres clés, vous pouvez utiliser l'option <code>_TM1CellValue_</code> pour mapper les valeurs de cube TM1 avec la table ODS. Vous <i>ne pouvez pas</i> utiliser l'option <code>_TM1CellValue_</code> si votre table ODS renferme plusieurs colonnes de chiffres clés.</p> <p>Si vous utilisez cette option, ne mappez pas la dimension de mesures TM1 sur la colonne de chiffres clés. A la place, cliquez sur la colonne Dimension <b>TM1</b> et sélectionnez <code>_TM1CellValue_</code>.</p> <p>Lorsque vous utilisez cette méthode alternative pour mapper les valeurs de cube TM1, la dimension de mesures TM1 ne doit <i>pas</i> être mappée à une colonne ODS.</p>

### Devise

Zone	Description
Devise SAP	Permet d'importer des données de devise dans un cube tridimensionnel, existant ou nouveau, sur votre serveur.
Afficher les noms techniques SAP	Pour employer les noms techniques, cochez cette case. Ne la cochez pas si vous souhaitez afficher les noms descriptifs.
Nom du cube	Entrez le nom d'un cube tridimensionnel existant ou un nouveau nom.
Devise à convertir	<p>Sélectionnez la devise initiale à importer depuis SAP.</p> <p>La liste des devises disponibles reflète les devises définies dans votre système SAP.</p>
Dimension cible	Indiquez la dimension TM1 devant recevoir les chaînes de devises SAP.
Devise cible	Sélectionnez la seconde devise à importer depuis SAP.

<b>Zone</b>	<b>Description</b>
Type de conversion	Sélectionnez la méthode de conversion à utiliser pour convertir la devise d'origine en devise cible.

### **TM1**

Utilise une dimension ou un cube TM1 comme source de données.

<b>Zone</b>	<b>Description</b>
Vue de cube Sous-ensemble de dimension	A l'aide du bouton Parcourir, sélectionnez une vue ou une dimension TM1 disponible à utiliser comme source de données. Cliquez sur le bouton Aperçu. Renseignez ensuite les zones des autres onglets.

### **IBM Cognos Package Connector**

Supprimé de la version 2.0.8 Indique que la source de données est un package IBM Cognos publié qui a été créé à partir d'une requête SAP.

### **Pack**

<b>Zone</b>		<b>Description</b>
Connexion	Définissez la connexion à cette source de données.	Définissez la connexion à cette source de données.
	Espace-noms d'authentification	Affiche tous les espaces-noms IBM Cognos créés qui sont actuellement disponibles.
	ID utilisateur	Mot de passe
Pack	Sélectionnez le pack	Cliquez sur le bouton Parcourir pour sélectionner un pack publié disponible.
	Sélectionnez le cube TM1 vers lequel effectuer le chargement	Si vous importez les données directement dans un cube TM1 existant, entrez le nom du cube ou sélectionnez-le dans la liste déroulante.
	Action de données, Action de cube, Activer le journal de cube	Consultez les descriptions de ces zones dans la section ci-dessus relative à l'InfoCube SAP.

### **Dimension**

<b>Zone</b>	<b>Description</b>	
Dimension	Pack	Identifiez le pack à utiliser pour cette dimension.

Zone	Description	
	Dimension à partir de laquelle effectuer le chargement	Identifiez la dimension à utiliser.
	Dimension TM1 à charger dans	Identifiez la dimension dans laquelle importer les données.
	Action de dimension TM1	Pour de plus amples détails, reportez-vous à la description de l'onglet Caractéristique.
	Récupérer les paramètres de sécurité	Utilisez la sécurité sur la dimension.
	Consolidation de premier niveau	Nom d'une consolidation de premier niveau pour la dimension TM1, avec tous les éléments importés en tant qu'enfants de la consolidation.
	Sélectionner des hiérarchies	Sélectionnez et mappez les hiérarchies à inclure dans l'importation, puis définissez leur mappage dans la nouvelle dimension TM1. Pour plus d'informations, reportez-vous à la documentation IBM Cognos <i>TM1 TurboIntegrator</i> .
	Sélectionner des attributs	Sélectionnez les attributs à utiliser et définissez le mappage. Pour plus d'informations, reportez-vous à la documentation IBM Cognos <i>TM1 TurboIntegrator</i> .

### Aucun

Permet d'ajouter à un processus un prologue défini par l'utilisateur.

Si la source de données d'un processus est Aucune, TurboIntegrator exécute immédiatement la procédure d'Épilogue une fois le traitement de Prologue terminé.

**Remarque :** Lorsque la source de données d'un processus est définie à Aucune, les procédures de métadonnées et de données sont ignorées. Dans ce cas, tous les scripts du processus doivent être créés soit dans la procédure de prologue, soit dans la procédure d'épilogue.

### Aperçu de grille

L'aperçu de grille affiche les dix premiers enregistrements de votre source de données. Servez-vous de cette grille pour confirmer que la source est correcte et pour vous aider à déterminer la structure des enregistrements.

Si vous changez la source de données, cliquez sur Aperçu de nouveau pour actualiser l'affichage de la grille.

### Onglet Variables

L'onglet Variables comprend une grille et deux boutons.

## Grille

Servez-vous de la grille Variables pour affecter les variables et identifier le contenu de chaque colonne de votre source de données. La grille Variables comprend les colonnes suivantes.

En colonnes	Description
ID de colonne	Liste chaque zone ou colonne unique identifiés dans la source de données. Les cellules de cette colonne ne sont pas modifiables.
Nom de variable	Contient une variable générée automatiquement pour chaque colonne de votre source de données. Toutes les variables générées sont nommées Vn, n étant égal à 0 pour la première colonne et incrémenté de 1 pour chaque colonne subséquente de la source.  Pour affecter vos propres variables, cliquez sur la cellule appropriée et saisissez la nouvelle variable.
Type de variable	Contient une liste pour chaque colonne de votre source de données. Utilisez cette liste pour spécifier si la variable est une chaîne ou un nombre.
Valeur d'exemple	Contient des valeurs exemple pour le premier enregistrement de la source. Les valeurs exemple vous aident à identifier le contenu de chaque colonne de la source. Les cellules de la colonne Valeur exemple ne sont pas modifiables.
Contenu	Contient une liste pour chaque colonne de votre source de données. Servez-vous de cette liste pour spécifier le type de valeur contenu dans chaque colonne de la source.
Formule	Cette colonne est grisée pour toutes les zones de la source et ne devient disponible que quand vous créez une nouvelle variable.  Quand vous créez une nouvelle variable, double-cliquez sur la cellule Formule associée pour ouvrir la boîte de dialogue Traiter la formule de variable qui permet de définir une formule pour la variable.

## Boutons

Bouton	Description
Nouvelle variable	Cliquez pour créer une nouvelle variable.
Supprimer	Cliquez pour supprimer une variable définie par l'utilisateur.

### Formule de variables de processus

La boîte de dialogue Formule de variables de processus affiche et permet l'édition de formules utilisées dans un processus TurboIntegrator. Si une formule existe et que vous cliquez sur l'onglet Formule, la formule actuellement définie est affichée dans la fenêtre Formule. Cliquez sur Nouvelle variable pour définir une nouvelle formule.

Option	Description
Formule	La formule telle que saisie s'affiche dans cette fenêtre. Cette fenêtre est mise à jour au fur et à mesure que vous entrez le texte de la formule.
Destination	Sélectionnez l'emplacement de cette formule en fonction de vos besoins de programmation pour ce processus. Sélectionnez Données pour placer la formule dans la section Données du processus TurboIntegrator. Sélectionnez Métadonnées pour la positionner dans la section Métadonnées. L'option Les deux positionne la formule aux deux emplacements.
Evaluer	Cliquez ici pour valider la formule.
Valeur exemple	Lorsque la formule est évaluée, des informations sur la formule sont affichées ici. Par exemple, <i>Ligne 1 : Erreur de syntaxe avant ou après : \n (fin de ligne) point-virgule manquant.</i>
Afficher automatiquement quand le nom de la variable change	Cliquez ici pour afficher cette boîte de dialogue si le nom de la variable a été modifié. Si cette zone n'est pas cochée, vous devez l'invoquer manuellement en cliquant sur la zone Formule dans l'onglet Variables.

### Onglet Mappages

L'onglet Mappages permet de définir la façon de mapper les données source à des cubes, dimensions, données, consolidations et attributs de la base de données TM1.

L'onglet Mappages est constitué d'une série de sous-onglets contenant chacun des options permettant de mapper les variables de la source de données à des structures de métadonnées TM1 existantes. Les sous-onglets disponibles varient suivant le type des valeurs contenues dans la source de données, comme spécifié dans la colonne Contenu de l'onglet Variables.

L'onglet Mappages contient les sous-onglets suivants :

#### Cube

Le sous-onglet Cube permet de spécifier la manière dont TurboIntegrator mappe les données importées aux cubes TM1. Le sous-onglet Cube contient les options suivantes :

Option	Description
Action de cube	Sélectionnez une option pour créer, mettre à jour, recréer ou n'appliquer aucune action à un cube.

Option	Description
Nom du cube	Spécifiez le cube auquel l'action s'applique. Si vous créez un nouveau cube, tapez son nom dans la zone de saisie. Sinon, sélectionnez un cube dans la liste.
Remplir la portion de zéros	Cette option devient disponible quand vous sélectionnez l'action Mettre un cube à jour. Sélectionnez cette case si vous voulez définir toutes les données individuelles d'une vue à zéro.
Nom de la vue	Cette option devient disponible quand vous sélectionnez les options <b>Mettre un cube à jour</b> et Remplir la portion de zéros. Sélectionnez ou définissez la vue comprenant les données individuelles que vous voulez définir à zéro.
Action de données	Sélectionnez une option pour déterminer comment les données traitées sont stockées dans le cube. <b>Stocker les valeurs</b> remplace les valeurs de cube existantes par les valeurs importées par le processus. <b>Accumuler les valeurs</b> ajoute les valeurs importées par le processus dans les valeurs de cube existantes.
Activer le journal de cube	Cochez cette case pour écrire les modifications du cube dans le fichier Tm1s.log. Désactivez cette case pour traiter les cubes sans écrire les modifications dans le fichier Tm1s.log.

### Dimensions

Utilisez le sous-onglet Dimensions pour mapper les variables d'élément sur les éléments de dimension.

Ce sous-onglet inclut une grille permettant de mapper des variables individuelles à des dimensions dans la base de données TM1. Cette grille comprend les colonnes suivantes :

En colonnes	Description
Variable d'élément	Contient le nom de chaque variable pour laquelle vous avez spécifié une valeur Contenu pour l'élément. La valeur Contenu est spécifiée sur l'onglet Variables.  Cette colonne contient aussi l'étiquette (Variables de données) pour toute variable ayant une valeur Contenu de Données.

En colonnes	Description
Valeur d'exemple	Une valeur exemple pour le premier enregistrement de votre source de données. Servez-vous de cette valeur pour vous faciliter l'identification de la dimension sur laquelle mappe la variable d'élément.
Dimension	Répertorie toutes les dimensions disponibles sur le serveur. Sélectionnez la dimension sur laquelle mappe la variable d'élément.  Pour mapper la variable d'élément sur une nouvelle dimension, tapez le nom de la nouvelle dimension dans la zone de saisie.
Ordre du cube	Cette option devient disponible quand l'Action du cube est Créer.  Indiquez l'ordre de chaque dimension dans le cube que vous créez.
Action	Répertorie les actions de dimension disponibles. Sélectionnez une action.  Pour créer une nouvelle dimension, vous devez spécifier une action <b>Créer</b> .
Type d'élément	Sélectionnez un type d'élément pour la variable, <b>Numérique</b> ou <b>Chaîne</b> .
Ordre des éléments	Sélectionnez une option pour ordonner les éléments dans toute dimension que vous créez ou actualisez. Il existe quatre ordres de tri :  Entrée - Trie les éléments dans l'ordre où ils ont été créés dans la dimension.  Nom - Trie les éléments en ordre alphabétique, soit croissant, soit décroissant.  Niveau - Trie les éléments par niveau dans la hiérarchie, soit croissant, soit décroissant.  Hiérarchie - Trie les éléments tels qu'ils sont dans la hiérarchie de dimension.

### Données

Servez-vous du sous-onglet Données pour mapper les variables de données sur des éléments spécifiques.

Ce sous-onglet inclut une grille permettant de mapper des variables individuelles à des éléments de la base de données TM1. Cette grille comprend les colonnes suivantes :



En colonnes	Description
Variable de données	Contient le nom de chaque variable pour laquelle vous avez spécifié la valeur de Contenu Données. La valeur Contenu est spécifiée sur l'onglet Variables.
Élément	Cliquez sur le bouton avec la flèche vers la droite pour ouvrir l'Editeur de sous-ensemble dans lequel vous sélectionnez l'élément sur lequel la variable est mappée.  Pour mapper la variable sur un nouvel élément, tapez le nom de l'élément dans la zone de saisie.
Type d'élément	Sélectionnez un type d'élément.
Valeur d'exemple	Une valeur exemple pour le premier enregistrement de votre source de données. Servez-vous de cette valeur pour vous faciliter l'identification de l'élément sur lequel mappe la variable de données.

### Consolidations

Servez-vous du sous-onglet Consolidations pour mapper des enfants sur les éléments consolidés.

Ce sous-onglet inclut une grille permettant de mapper des variables individuelles à des dimensions dans la base de données TM1. Cette grille comprend les colonnes suivantes :

En colonnes	Description
Variable Variable	Contient le nom de chaque variable pour laquelle vous avez spécifié la valeur de Contenu Consolidation. La valeur Contenu est spécifiée sur l'onglet Variables.
Dimension	Liste des dimensions sur lesquelles la consolidation peut être mappée.
Variable enfant	Liste de variables dans laquelle vous sélectionnez l'enfant immédiat de la consolidation.
Pondération	Affecte une pondération à la variable enfant spécifiée.
Valeur d'exemple	Une valeur exemple pour le premier enregistrement de votre source de données. Servez-vous de cette valeur pour vous faciliter l'identification de l'élément sur lequel mapper la consolidation.

En colonnes	Description
Ordre des éléments	<p>Sélectionnez une option pour ordonner les éléments dans toute consolidation que vous créez ou actualisez. Il existe quatre ordres de tri :</p> <p>Entrée - Trie les éléments dans l'ordre où ils ont été créés dans la dimension.</p> <p>Nom - Trie les éléments en ordre alphabétique, soit croissant, soit décroissant.</p> <p>Niveau - Trie les éléments par niveau dans la hiérarchie, soit croissant, soit décroissant.</p> <p>Hiérarchie - Trie les éléments tels qu'ils sont dans la hiérarchie de dimension.</p>

### Attributs

Servez-vous du sous-onglet Attributs pour mapper les variables d'attribut sur des attributs spécifiques.

Ce sous-onglet inclut une grille permettant de mapper des variables individuelles à des dimensions dans la base de données TM1. Cette grille comprend les colonnes suivantes :

En colonnes	Description
Variable d'attribut	Contient le nom de chaque variable pour laquelle vous avez spécifié la valeur de Contenu Attribut. La valeur Contenu est spécifiée sur l'onglet Variables.
Valeur d'exemple	Affiche une valeur exemple de la source de données. Servez-vous de cet exemple pour vous aider à mapper l'attribut.
Dimension	Répertorie toutes les dimensions disponibles sur le serveur. Sélectionnez la dimension à laquelle l'attribut s'applique.
Variable d'élément	Liste des variables d'élément. Sélectionnez la variable pour l'élément auquel la variable d'attribut s'applique.
Attribut	Liste des attributs sur lesquels la variable peut être mappée. Sélectionnez l'attribut approprié dans cette liste.
Action	Sélectionnez soit de Créer un nouvel attribut, soit de Mettre à jour un attribut existant.
Type d'attribut	Identifie le type de l'attribut sélectionné dans la colonne Attribut.

### Onglet Expert

L'onglet Expert contient plusieurs sous-onglets affichant les instructions générées par TM1 à partir des options que vous sélectionnez ailleurs dans l'Editeur TurboIntegrator. L'onglet Expert contient aussi un sous-onglet qui permet de définir les paramètres de processus.

## Paramètres

Élément	Description
Insérer	Cliquez pour insérer un nouveau paramètre.
Supprimer	Cliquez pour supprimer un paramètre sélectionné.
Paramètres	Tapez un nom pour chaque nouveau paramètre.
Type	Sélectionnez un type pour chaque paramètre.
Valeur par défaut	Entrez une valeur à utiliser comme valeur par défaut pour ce paramètre lors de l'exécution du processus TurboIntegrator.
Question à montrer	Entrez une invite à utiliser pour ce paramètre lors de l'exécution du processus TurboIntegrator.

## Prologue

Élément	Description
Zone de texte d'instruction	<p>Affiche des instructions générées définissant une série d'actions à exécuter avant le traitement de la source de données.</p> <p>Vous pouvez améliorer un processus en créant des instructions supplémentaires avec les fonctions de règles ou les fonctions TurboIntegrator.</p>
Bouton Aller à la ligne	Cliquez sur ce bouton, entrez la ligne à laquelle vous voulez aller puis cliquez sur OK pour aller directement à une ligne de code dans la zone de texte d'instruction.

## Métadonnées

Élément	Description
Zone de texte d'instruction	<p>Affiche des instructions générées définissant une série d'actions à exécuter sur les métadonnées TM1 avant le traitement de la source de données.</p> <p>Vous pouvez améliorer un processus en créant des instructions supplémentaires avec les fonctions de règles ou les fonctions TurboIntegrator.</p>
Bouton Aller à la ligne	Cliquez sur ce bouton, entrez la ligne à laquelle vous voulez aller puis cliquez sur OK pour aller directement à une ligne de code dans la zone de texte d'instruction.

## Données

Élément	Description
Zone de texte d'instruction	Affiche des instructions générées définissant une série d'actions devant être exécutées avant le traitement de la source de données.  Vous pouvez améliorer un processus en créant des instructions supplémentaires avec les fonctions de règles ou les fonctions TurboIntegrator.
Bouton Aller à la ligne	Cliquez sur ce bouton, entrez la ligne à laquelle vous voulez aller puis cliquez sur OK pour aller directement à une ligne de code dans la zone de texte d'instruction.

## Epilogue

Élément	Description
Zone de texte d'instruction	Affiche des instructions générées définissant une série d'actions devant être exécutées avant le traitement de la source de données.  Vous pouvez améliorer un processus en créant des instructions supplémentaires avec les fonctions de règles ou les fonctions TurboIntegrator.
Bouton Aller à la ligne	Cliquez sur ce bouton, entrez la ligne de destination, puis cliquez sur <b>OK</b> pour accéder directement à une ligne de code dans la zone de texte d'instruction.

## Onglet Planning

Utilisez cet onglet pour planifier un processus à exécuter à intervalles réguliers.

Élément	Description
Planifier ce processus comme tâche nommée	Cochez cette case pour exécuter ce processus en tant que tâche à intervalles réguliers. Par défaut, la tâche porte le même nom que le processus. Si vous voulez affecter un nom différent à la tâche, tapez ce nom dans la zone de saisie.
Date et heure de début de la tâche	Sélectionnez une date du calendrier pour spécifier une heure de début dans la zone Heure.
Fréquence d'exécution de la tâche	Remplissez les zones appropriées pour définir l'intervalle dans lequel la tâche doit être exécutée.

## Fenêtre Extraction de vue

Servez-vous de la fenêtre Extraction de vue pour créer une vue comprenant seulement les valeurs répondant aux critères définis par l'utilisateur ou pour définir une vue à exporter.


## Ignorer les paramètres



Paramètre	Description
Ignorer les valeurs calculées	Activez cette option pour ignorer les valeurs venant de consolidations lors de l'extraction de vue. Désactivez cette option pour inclure les valeurs venant de consolidations lors de l'extraction de vue. Par défaut, l'option est désactivée.
Ignorer les valeurs calculées par règle	Activez cette option pour ignorer les valeurs dérivées de règles lors de l'extraction de vue. Désactivez cette option pour inclure les valeurs dérivées de règles lors de l'extraction de vue. Par défaut, l'option est désactivée.
Ignorer les valeurs nulles/vides	Activez cette option pour ignorer les valeurs nulles ou vides lors de l'extraction de vue. Désactivez cette option pour inclure les valeurs nulles ou vides lors de l'extraction de vue. La valeur par défaut est option activée.

## Paramètres de plage

Paramètre	Description
Opérateur	Sélectionnez un opérateur définissant les valeurs à extraire.
Limites numériques	Entrez une valeur numérique pour la ou les variable(s) dans la zone Opérateur.
Limites du texte	Entrez une valeur de chaîne pour la ou les variable(s) dans la zone Opérateur.

## Sélection d'éléments de dimension

Pour chaque dimension, cliquez sur le bouton **Sous-ensemble**  et sélectionnez les éléments ou le sous-ensemble définissant les paramètres d'extraction de la vue.

Si la vue pour laquelle vous créez une extraction comporte plus de 16 dimensions, cliquez sur  pour revenir à la page contenant les 16 dimensions précédentes ou sur le bouton  pour accéder à celle contenant les 16 dimensions suivantes.

## Boîte de dialogue Styles de vue

La boîte de dialogue Styles de vue permet d'appliquer des styles Excel à la vue de cube TM1 dans le Navigateur de feuille de calcul. Cette boîte de dialogue contient plusieurs listes permettant d'appliquer un style Excel existant à une plage de cellules, ainsi que des boutons permettant de modifier ou créer des styles.

Élément	Description
Arrière-plan	Sélectionnez un style dans la liste pour l'appliquer à l'arrière-plan du Navigateur de feuille de calcul.
Cellules de données	Sélectionnez un style dans la liste pour l'appliquer aux cellules de données. Le style des Cellules de données prime sur le style de l'Arrière-plan.
Cellules d'en-tête de ligne	Sélectionnez un style dans la liste pour l'appliquer aux cellules d'en-tête de ligne. Le style des Cellules d'en-tête de ligne prime sur le style de l'Arrière-plan.
Cellules d'en-tête de colonne	Sélectionnez un style dans la liste pour l'appliquer aux cellules d'en-tête de colonne. Le style des Cellules d'en-tête de colonne prime sur le style de l'Arrière-plan.
Boutons Editer un style	Cliquez sur le bouton Editer un style approprié pour modifier ou créer des styles pour la plage associée dans le Navigateur de feuille de calcul.
Figeage des sous-fenêtres	<p>Basculez cette option pour figer ou défiger les sous-fenêtres dans le Navigateur de feuille de calcul.</p> <p>Quand cette option est basculée vers on, les noms de ligne d'élément restent visibles quand vous faites défiler la vue horizontalement et les éléments en colonne restent visibles quand vous faites défiler la vue verticalement.</p> <p>Quand cette option est basculée vers off, les noms d'éléments en ligne et en colonne se déplacent avec les valeurs de cube quand vous faites défiler une vue.</p>

## Chapitre 2. Fonctions de règles

Cette section contient une liste complète de toutes les fonctions de règles d'IBM TM1. Vous pouvez utiliser ces fonctions lorsque vous créez des règles TM1.

### Opérateurs arithmétiques dans les règles TM1

Les opérateurs arithmétiques suivants peuvent être utilisés pour créer des règles TM1.

Opérateur	Signification
+ (signe plus)	Addition
- (signe moins)	Soustraction
* (astérisque)	Multiplication
/ (barre oblique)	La division par zéro avec cet opérateur renvoie une valeur non définie.
\ (barre oblique inverse)	La division par zéro avec cet opérateur renvoie zéro.
^ (circonflexe)	Élévation à une puissance

### Opérateurs de comparaison dans les règles TM1

Les opérateurs de comparaison permettent de comparer des valeurs dans la partie formule d'une instruction de calcul de règle.

Pour comparer deux valeurs de chaîne, insérez le symbole @ avant l'opérateur de comparaison. Par exemple, IF ( 'A' @= 'B' , 0 , 1 ) génère le nombre 1.

Opérateur	Signification
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
=	Egal à
<>	Différent de

### Opérateurs logiques dans les règles TM1

Vous pouvez combiner des expressions dans une instruction de calcul de règle à l'aide d'opérateurs logiques.

Opérateur	Signification	Exemple
& (perluette)	ET	(Valeur1 > 5) & (Valeur1 < 10) Renvoie TRUE si la valeur est supérieure à 5 et inférieure à 10.
% (signe de pourcentage)	OU	(Valeur1 > 10) % (Valeur1 < 5) Renvoie TRUE si la valeur est supérieure à 10 ou inférieure à 5.
~ (tilde)	PAS	~(Valeur1 > 5) Est équivalent à (Valeur1 <= 5)

## Fonctions de règles d'attribut

Fonctions de règles qui utilisent des attributs.

### ATTRN

ATTRN renvoie un attribut numérique pour un élément spécifié d'une dimension.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
ATTRN(dimension, element, attribute)
```

Argument	Description
dimension	Nom de dimension valide.
element	Élément de la dimension.



Argument	Description
attribute	<p>Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.</p> <p><b>Remarque :</b> Lorsque cette fonction est utilisée dans une instruction conditionnelle (IF), l'instruction est la partie contenant la condition, pas l'ensemble du bloc conditionnel. En cas d'erreur mineure, l'exécution se poursuit avec l'instruction suivante. Une fois l'erreur mineure traitée, le traitement TI ne peut pas savoir qu'il s'agissait d'une instruction conditionnelle. Il considère donc la ligne suivante comme prochaine instruction et non la ligne qui suit ENDIF (fin du bloc conditionnel).</p> <p>Pour éviter cette situation, utilisez des variables pour chaque opération susceptible de rencontrer une erreur mineure, puis utilisez les variables dans l'instruction conditionnelle. Par exemple :</p> <pre>V1 = CELLGETN('PNLCube', 'fred', 'argentina', 'Sales', 'Jan'); IF(V1 = 454);ASCIIOUTPUT ('bug.txt', 'if logic not working properly'); ENDIF;</pre>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Engine Size de l'élément L Series 1.8L Sedan dans la dimension Model.

```
ATTRN('Model', 'L Series 1.8L Sedan', 'Engine Size')
```

## ATTRS

ATTRS renvoie un attribut de chaîne pour un élément spécifié d'une dimension.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ATTRS(dimension, element, attribute)
```

Argument	Description
dimension	Nom de dimension valide.
element	Élément de la dimension.
attribute	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.

## Exemple

Dans cet exemple, la fonction renvoie la valeur chaîne de l'attribut Currency de l'élément 10100 dans la dimension plan\_business\_unit.

```
ATTRS('plan_business_unit', '10100', 'Currency')
```

## CubeATTRN

CubeATTRN renvoie un attribut numérique d'un cube donné.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
CubeATTRN(CubeName, AttrName);
```

Argument	Description
CubeName	Nom de cube valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide du cube.

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code du cube Product.

```
CubeATTRN('Product', 'Accounting_Code');
```

## CubeATTRS

CubeATTRS renvoie un attribut de type chaîne d'un cube donné.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
CubeATTRS(CubeName, AttrName);
```

Argument	Description
CubeName	Nom de cube valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide du cube.

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner du cube Product.

```
CubeATTRS('Product', 'Owner');
```

## DimensionATTRN

DimensionATTRN renvoie un attribut numérique d'une dimension donnée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
DimensionATTRN(DimName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code de la dimension Plan\_Business\_Unit.

```
DimensionATTRN('Plan_Business_Unit', 'Accounting_Code');
```

## DimensionATTRS

DimensionATTRS renvoie un attribut de type chaîne d'une dimension donnée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
DimensionATTRS(DimName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Manager de la dimension Plan\_Business\_Unit.

```
DimensionATTRS('Plan_Business_Unit', 'Manager');
```

## ElementAttrN

ElementAttrN renvoie un attribut numérique pour un élément spécifié d'une dimension.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

```
ElementAttrN(dimension, hierarchy, element, attribute)
```

Argument	Description
dimension	Nom de dimension valide.

Argument	Description
hierarchy	Nom de la hiérarchie dans la dimension.
element	Élément de la dimension.
attribute	<p>Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.</p> <p><b>Remarque :</b> Lorsque cette fonction est utilisée dans une instruction conditionnelle (IF), l'instruction est la partie contenant la condition, pas l'ensemble du bloc conditionnel. En cas d'erreur mineure, l'exécution se poursuit avec l'instruction suivante. Une fois l'erreur mineure traitée, le traitement TI ne peut pas savoir qu'il s'agissait d'une instruction conditionnelle. Il considère donc la ligne suivante comme prochaine instruction et non la ligne qui suit ENDIF (fin du bloc conditionnel).</p> <p>Pour éviter cette situation, utilisez des variables pour chaque opération susceptible de rencontrer une erreur mineure, puis utilisez les variables dans l'instruction conditionnelle. Par exemple :</p> <pre>V1 = CELLGETN('PNLCube', 'fred', 'argentina', 'Sales', 'Jan'); IF(V1 = 454);ASCIIOUTPUT ('bug.txt', 'if logic not working properly'); ENDIF;</pre>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Engine Size de l'élément L Series 1.8L Sedan dans la hiérarchie Automobile de la dimension Model.

```
ElementAttrN('Model', 'Automobile', 'L Series 1.8L Sedan', 'Engine Size')
```

### ElementAttrS

ElementAttrS renvoie un attribut de chaîne pour un élément spécifié d'une dimension.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementAttrS(dimension, hierarchy, element, attribute)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Élément de la dimension.

Argument	Description
attribute	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.

### Exemple

Dans cet exemple, la fonction renvoie la valeur chaîne de l'attribut Currency de l'élément 10100 dans la hiérarchie expense de la dimension plan\_business\_unit.

```
ElementAttrS('plan_business_unit', 'expense', '10100', 'Currency')
```

## Fonctions de règles de calcul de consolidation

Les fonctions ConsolidatedMax; ConsolidatedMin; ConsolidatedAvg; ConsolidatedCount; et ConsolidatedCountUnique effectuent des calculs mathématiques sur les consolidations.

### ConsolidatedAvg

Cette fonction calcule la valeur moyenne dans une consolidation et renvoie une valeur unique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
ConsolidatedAvg(flag-value, cube-name, element_1, element_2,... );
```

#### Arguments

##### flag-value

La valeur d'indicateur correspond à la somme des valeurs suivantes :

- 1 : utilise la pondération de consolidation lors du calcul de la valeur (inclure les zéros dans le calcul). Si cette valeur n'est pas activée, la valeur brute de l'élément consolidé est utilisée. Les conditions suivantes peuvent avoir une incidence sur l'inclusion des zéros dans le calcul :
  - Si zéro est spécifié comme pondération de certains éléments consolidés, alors le paramètre de configuration : ZeroWeightOptimization=F doit être défini pour que ces éléments soient factorisés dans le calcul des fonctions. Sans ce paramètre de configuration, les éléments pour lesquels la pondération est zéro sont éliminés de la liste de consolidation, et ne sont donc pas traités lors de la découverte d'éléments comme la valeur minimale ou maximale.
  - Si vous souhaitez que les cellules comportant la valeur zéro soient prises en compte dans la moyenne, définissez la variable UNDEFVALS pour le cube de sorte que cette valeur soit stockée dans la cellule comme n'importe quelle autre valeur différente de zéro.
  - Si une règle de cube inclut une variable SKIPCHECK, les zéros sont ignorés même si flag-value a la valeur 1. Supprimez SKIPCHECK de la règle afin d'inclure les zéros dans la moyenne consolidée.
- 2 : Ignore les valeurs nulles. Si cette valeur est définie, les valeurs zéro ne sont pas incluses dans le calcul.

##### cube-name

Nom du cube dans lequel les valeurs résident.

Si la fonction s'exécute dans le cadre d'une règle de cube, et NON dans le cadre d'un processus TurboIntegrator, l'argument cube-name peut être indiqué sous la forme d'une chaîne vide pour désigner le cube en cours. Cela signifie que vous pouvez écrire une règle telle que `['Apr']=ConsolidatedAvg( 0, '', !actvsbud, '1 Quarter' );`

## element\_1, element\_2, ...

Noms des éléments de dimension définissant l'intersection du cube qui contient la valeur à récupérer.

Les arguments *élément\_1* à *élément\_n* doivent suivre un ordre précis. L'*élément\_1* doit être un élément de la première dimension du cube, l'*élément\_2* un élément de la deuxième, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

Dans un cube appelé Compte de résultat comportant trois dimensions (Régions, Temps et Compte de résultat), la dimension Compte de résultat contient un élément appelé Ventes brutes pour les chiffres de ventes globaux.

Pour calculer les ventes moyennes de toutes les régions en 2010, écrivez ce qui suit :

```
ConsolidatedAvg( 1, 'Income Statement', 'All Regions', '2010', 'Gross Sales' );
```

## ConsolidateChildren

Cette fonction force le calcul des valeurs consolidées en additionnant les enfants immédiats le long d'une dimension spécifique. La fonction ConsolidateChildren est utile lorsque des consolidations intermédiaires sont calculées par des règles et que souhaitez qu'une consolidation parente soit calculée en additionnant les consolidations intermédiaires plutôt que les valeurs de feuilles sous-jacentes.

Cette fonction est valide uniquement dans les règles TM1.

### Syntaxe

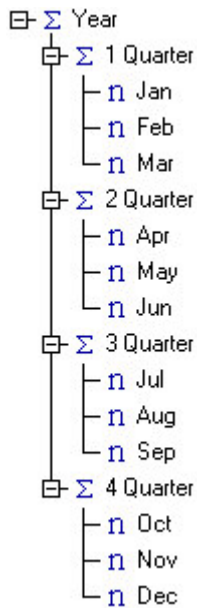
```
ConsolidateChildren(DimName1, DimName2, ...)
```

Argument	Description
DimName1, DimName2, ...	Noms des dimensions le long desquelles les consolidations seront effectuées.  Cette fonction requiert au moins un argument DimName et peut en accepter autant d'arguments DimName qu'il y a dimensions dans le cube pour lequel la règle est écrite.

### Exemple

Considérons un cube nommé Sales composé des dimensions ActVsBud, Region, Model, Account1 et Month.

Dans cet exemple, la dimension Month est définie comme suit :



Si aucune règle n'est en place pour ce cube, la valeur de la consolidation Year est calculée en additionnant toutes les valeurs de feuilles sous-jacentes, dans ce cas Jan à Dec. L'image suivante présente cette consolidation.

Actual		Denmark		S Series 1.8 L Sedan		Units										
month																
- Year	- 1 Quarter	Jan	Feb	Mar	- 2 Quarter	Apr	May	Jun	- 3 Quarter	Aug	Sep	Jul	- 4 Quarter	Nov	Dec	Oct
3156	770	223	331	216	906	311	250	345	757	222	197	338	723	199	212	312

Supposons maintenant que vous créez la règle suivante pour ce cube, qui indique que toutes les valeurs trimestrielles doivent être 1 :

```
[{'1 Quarter', '2 Quarter', '3 Quarter', '4 Quarter'}]=1;
```

Le résultat est le suivant :

Actual		Denmark		S Series 1.8 L Sedan		Units										
month																
- Year	- 1 Quarter	Jan	Feb	Mar	- 2 Quarter	Apr	May	Jun	- 3 Quarter	Aug	Sep	Jul	- 4 Quarter	Nov	Dec	Oct
3156	1	223	331	216	1	311	250	345	1	222	197	338	1	199	212	312

Dans l'image suivante, vous pouvez constater que les valeurs trimestrielles sont vraiment calculées par la règle mais la consolidation Year est toujours calculée en additionnant toutes les valeurs de feuilles sous-jacentes. S'il ne s'agit pas du chemin de calcul souhaité, vous pouvez utiliser la fonction ConsolidateChildren pour forcer TM1 à calculer la consolidation Year en additionnant ses enfants immédiats, spécifiquement 1 Quarter, 2 Quarter, 3 Quarter et 4 Quarter.

```
['Year']=ConsolidateChildren('Month');[{'1 Quarter', '2 Quarter', '3 Quarter', '4 Quarter'}]=1;
```

Dans la règle, l'instruction ['Year']=ConsolidateChildren('Month') indique que la consolidation Year doit être calculée en totalisant les enfants immédiats de Year dans la dimension Month.

L'image qui suit présente le résultat de l'instruction ['Year']=ConsolidateChildren('Month') :

Actual		Denmark		S Series 1.8 L Sedan		Units										
month																
- Year	- 1 Quarter	Jan	Feb	Mar	- 2 Quarter	Apr	May	Jun	- 3 Quarter	Aug	Sep	Jul	- 4 Quarter	Nov	Dec	Oct
4	1	223	331	216	1	311	250	345	1	222	197	338	1	199	212	312

Notez que la consolidation Year est maintenant calculée en totalisant ses enfants immédiats.

Il est important de retenir que, pour une consolidation donnée, la fonction ConsolidateChildren s'applique seulement aux enfants *immédiats* de la consolidation.

La fonction ConsolidateChildren peut aussi être utilisée pour spécifier le mode de calcul des consolidations dans plusieurs dimensions, comme dans l'exemple suivant :

Argument	Description
['World','Year']= ConsolidateChildren('Region','Month')	Cette instruction applique la fonction ConsolidateChildren à la fois aux consolidations World et Year. Dans ce cas, World est calculé en totalisant tous ses enfants immédiats dans la dimension Region, alors que Year est calculé en totalisant tous ses enfants immédiats dans la dimension Month.

## ConsolidatedCount

Cette fonction comptabilise la valeur dans une consolidation et renvoie une valeur unique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ConsolidatedCount(flag-value, cube-name, element_1, element_2,... );
```

### Arguments

#### flag-value

La valeur d'indicateur correspond à la somme des valeurs suivantes :

- 1 : utilise la pondération de consolidation lors du calcul de la valeur. Si cette option n'est pas activée, la valeur brute de l'élément consolidé est utilisée.
- 2 : ignore les valeurs nulles. Si cette option est activée, les valeurs nulles ne sont pas utilisées par ces fonctions.

**Remarque :** Si zéro est spécifié comme pondération de certains éléments consolidés, alors le paramètre de configuration ZeroWeightOptimization=F doit être défini pour que ces éléments soient factorisés dans le calcul des fonctions. Sans ce paramètre de configuration, les éléments pour lesquels la pondération est zéro sont éliminés de la liste de consolidation, et ne sont donc pas traités lors de la découverte de choses sur la ligne de la valeur minimale ou maximale.

#### cube-name

Nom du cube dans lequel les valeurs résident.

Si la fonction s'exécute dans le cadre d'une règle de cube, et NON dans le cadre d'un processus TurboIntegrator, l'argument cube-name peut être indiqué sous la forme d'une chaîne vide pour désigner le cube en cours. Cela signifie que vous pouvez écrire une règle telle que la suivante :  
['Apr']=ConsolidatedCount( 1, '', !actvsbud, '1 Quarter' );

#### element\_1, element\_2, ...

Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.

Les arguments element\_1 à element\_n doivent suivre un ordre précis. L'argument element\_1 doit être un élément de la première dimension du cube, element\_2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.



## ConsolidatedCountUnique

Cette fonction comptabilise le nombre d'éléments uniques pour lesquels il existe des points de données pour la consolidation spécifiée. Les éléments uniques sont comptabilisés avec une dimension de la cellule consolidée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ConsolidatedCountUnique( 0, unique-along-dimension-name, cube-name, elem_1, elem_2, . . . );
```

### Arguments

#### flag word

Spécifiez 0 (zéro).

**Remarque :** Si zéro est spécifié comme pondération de certains éléments consolidés, alors le paramètre de configuration ZeroWeightOptimization=F doit être défini pour que ces éléments soient factorisés dans le calcul des fonctions. Sans ce paramètre de configuration, les éléments pour lesquels la pondération est zéro sont éliminés de la liste de consolidation, et ne sont donc pas traités lors de la découverte d'éléments comme la valeur minimale ou maximale.

#### unique-along-dimension-name

La dimension associée aux entrées d'éléments uniques pour lesquelles il existe des données de type real doivent être comptabilisées.

#### cube-name

Nom du cube dans lequel les valeurs résident.

Si la fonction s'exécute dans le cadre d'une règle de cube, et NON dans le cadre d'un processus TurboIntegrator, l'argument cube-name peut être indiqué sous la forme d'une chaîne vide pour désigner le cube en cours.

#### element\_1, element\_2, ...

Noms des éléments de dimension définissant l'intersection du cube contenant la valeur consolidée à traiter.

Les arguments element\_1 à element\_n doivent suivre un ordre précis. L'argument element\_1 doit être un élément de la première dimension du cube, element\_2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

Dans un cube appelé Income Statement comportant trois dimensions (Regions, Time et Income Statement), la dimension Income Statement contient un élément, appelé Gross Sales, pour les chiffres de ventes globaux. Pour comptabiliser le nombre de régions ayant réalisé des ventes brutes au cours de l'année 2010, écrivez ce qui suit :

```
ConsolidatedCountUnique( 0, 'Regions', 'Income Statement', 'All Regions', '2010', 'Gross Sales' );
```

## ConsolidatedMax

Cette fonction calcule la valeur maximale dans une consolidation et renvoie une valeur unique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ConsolidatedMax(flag-value, cube-name, element_1, element_2,... );
```

## Arguments

### flag-value

La valeur d'indicateur correspond à la somme des valeurs suivantes :

1 : utilise la pondération de consolidation lors du calcul de la valeur. Si cette option n'est pas activée, la valeur brute de l'élément consolidé est utilisée.

2 : ignore les valeurs nulles. Si cette option est activée, les valeurs nulles ne sont pas utilisées par ces fonctions.

**Remarque :** Si zéro est spécifié comme pondération de certains éléments consolidés, alors le paramètre de configuration ZeroWeightOptimization=F doit être défini pour que ces éléments soient factorisés dans le calcul des fonctions. Sans ce paramètre de configuration, les éléments pour lesquels la pondération est zéro sont éliminés de la liste de consolidation, et ne sont donc pas traités lors de la découverte de choses sur la ligne de la valeur minimale ou maximale.

### cube-name

Nom du cube dans lequel les valeurs résident.

Si la fonction s'exécute dans le cadre d'une règle de cube, et NON dans le cadre d'un processus TurboIntegrator, l'argument cube-name peut être indiqué sous la forme d'une chaîne vide pour désigner le cube en cours. Cela signifie que vous pouvez écrire une règle telle que la suivante :  
['Apr']=ConsolidatedMax( 1, '', !actvsbud, '1 Quarter' );

### element\_1, element\_2, ...

Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.

Les arguments element\_1 à element\_n doivent suivre un ordre précis. L'argument element\_1 doit être un élément de la première dimension du cube, element\_2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

## Exemple

Prenons l'exemple d'un cube, appelé Income Statement (Compte de résultat), qui comporte trois dimensions ("Area" (Régions), "Time" (Temps) et "Income Statement" (Compte de résultat)). La dimension Income Statement contient un élément "Gross Sales" (Ventes brutes) pour les chiffres de ventes globaux.

Pour calculer les ventes moyennes de toutes les régions en 2010, écrivez ce qui suit :

```
ConsolidatedMax( 1, 'Income Statement', 'All Regions', '2010', 'Gross Sales' );
```

## ConsolidatedMin

Cette fonction calcule la valeur minimale dans une consolidation et renvoie une valeur unique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
ConsolidatedMin(flag-value, cube-name, element_1, element_2,... );
```

## Arguments

### flag-value

La valeur d'indicateur correspond à la somme des valeurs suivantes :

1 : utilise la pondération de consolidation lors du calcul de la valeur. Si cette option n'est pas activée, la valeur brute de l'élément consolidé est utilisée.

2 : ignore les valeurs nulles. Si cette option est activée, les valeurs nulles ne sont pas utilisées par ces fonctions.

**Remarque :** Si zéro est spécifié comme pondération de certains éléments consolidés, alors le paramètre de configuration `ZeroWeightOptimization=F` doit être défini pour que ces éléments soient factorisés dans le calcul des fonctions. Sans ce paramètre de configuration, les éléments pour lesquels la pondération est zéro sont éliminés de la liste de consolidation, et ne sont donc pas traités lors de la découverte de choses sur la ligne de la valeur minimale ou maximale.

### cube-name

Nom du cube dans lequel les valeurs résident.

Si la fonction s'exécute dans le cadre d'une règle de cube, et NON dans le cadre d'un processus TurboIntegrator, l'argument `cube-name` peut être indiqué sous la forme d'une chaîne vide pour désigner le cube en cours. Cela signifie que vous pouvez écrire une règle telle que la suivante :  
`['Apr']=ConsolidatedMin( 1, '', !actvsbud, '1 Quarter' );`

### element\_1, element\_2, ...

Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.

Les arguments `element_1` à `element_n` doivent suivre un ordre précis. L'argument `element_1` doit être un élément de la première dimension du cube, `element_2` un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

## Fonctions de règles pour données de cube

Fonctions de règles qui utilisent des données de cube.

### CellValueN

`CellValueN` renvoie la valeur numérique du ou des éléments spécifiés dans un cube. Il s'agit d'une fonction de règles TM1 qui est valide uniquement dans les règles TM1. L'emploi de cette fonction dans un processus TurboIntegrator génère une erreur.

Pour les dimensions en dehors des paramètres d'élément, les coordonnées sont extraites de la cible de règle (cellule extraite qui déclenche l'évaluation de la règle). La fonction se comporte de manière similaire à l'expression de référence intracube (par exemple, `['Measures':'Count']`), telle qu'elle est utilisée sur le côté droit de la règle.

Les paramètres d'élément peuvent être spécifiés dans n'importe quel ordre, et pour `CellValueN`, il est possible de spécifier plusieurs éléments de la même dimension (mais différentes hiérarchies de la dimension). Comme il n'est pas nécessaire que les éléments soient répertoriés dans l'ordre des dimensions du cube, tous les paramètres d'élément doivent être qualifiés par la dimension. Les paramètres d'élément des dimensions à plusieurs hiérarchies doivent également être qualifiés par la hiérarchie.

### Syntaxe

```
CellValueN(cube, element1,..., elementN);
```

Argument	Description
cube	Nom du cube.
elementN	Nom d'élément qui définit la cellule. Au moins un élément doit être spécifié.

## Exemple

```
CellValueS('ForecastCube', 'Products':'ProductsByChannel':'Channel2', 'Measures':'Count');
```

Cet exemple renvoie la valeur numérique de la cellule spécifiée. La dimension Products contient plusieurs hiérarchies alors que la dimension Measures n'en contient qu'une.

La référence intracube est restreinte aux paramètres littéraux, contrairement à CellValueN. Ce comportement est analogue à celui de la fonction de règles DB(). Les paramètres d'élément peuvent être spécifiés à l'aide d'expressions de chaîne. Par exemple, le paramètre d'élément Products ci-dessus peut être spécifié comme suit :

```
'Products' : 'ProductsByChannel' : AttrS( ... )
```

Contrairement à DB() et à l'expression de référence intracube, les paramètres d'élément CellValueN doivent être qualifiés par la dimension ou la dimension et la hiérarchie.

## CellValueS

CellValueS renvoie la valeur chaîne du ou des éléments spécifiés dans un cube. Il s'agit d'une fonction de règles TM1 qui est valide uniquement dans les règles TM1. L'emploi de cette fonction dans un processus TurboIntegrator génère une erreur.

Pour les dimensions en dehors des paramètres d'élément, les coordonnées sont extraites de la cible de règle (cellule extraite qui déclenche l'évaluation de la règle). La fonction se comporte de manière similaire à l'expression de référence intracube (par exemple, [ 'Measures':'Count' ]), telle qu'elle est utilisée sur le côté droit de la règle.

Les paramètres d'élément peuvent être spécifiés dans n'importe quel ordre, et pour CellValueS, il est possible de spécifier plusieurs éléments de la même dimension (mais différentes hiérarchies de la dimension). Comme il n'est pas nécessaire que les éléments soient répertoriés dans l'ordre des dimensions du cube, tous les paramètres d'élément doivent être qualifiés par la dimension. Les paramètres d'élément des dimensions à plusieurs hiérarchies doivent également être qualifiés par la hiérarchie.

## Syntaxe

```
CellValueS(cube, element1,..., elementN);
```

Argument	Description
cube	Nom du cube.
elementN	Nom d'élément qui définit la cellule. Au moins un élément doit être spécifié.

## Exemple

```
CellValueS('ForecastCube', 'Products':'ProductsByChannel':'Channel2', 'Measures':'Location');
```

Cet exemple renvoie la valeur de chaîne de la cellule spécifiée. La dimension Products contient plusieurs hiérarchies alors que la dimension Measures n'en contient qu'une.

La référence intracube est restreinte aux paramètres littéraux, contrairement à CellValueS. Ce comportement est analogue à celui de la fonction de règles DB(). Les paramètres d'élément peuvent être spécifiés à l'aide d'expressions de chaîne. Par exemple, le paramètre d'élément Products ci-dessus peut être spécifié comme suit :

```
'Products' : 'ProductsByChannel' : AttrS( ... )
```

Contrairement à DB() et à l'expression de référence intracube, les paramètres d'élément CellValueS doivent être qualifiés par la dimension ou la dimension et la hiérarchie.

## DB

DB renvoie une valeur depuis un cube d'une base de données TM1. DB renvoie une valeur numérique si la fonction est utilisée dans une expression numérique et une chaîne si elle est utilisée dans une expression chaîne.

DB est une fonction de règles TM1, valide dans les règles TM1 uniquement. L'emploi de cette fonction dans un processus TurboIntegrator produit une erreur.

### Syntaxe

```
DB(cube, e1, e2, [...e256])
```

### Paramètres

#### cube

Nom du cube depuis lequel extraire la valeur.

#### e1,...en

Noms d'élément de dimension qui définissent l'intersection où se trouve la valeur à récupérer.

Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite.

### Exemple

Dans cet exemple, Budget est le nom du cube et la fonction renvoie la valeur se trouvant à l'intersection de California, 15" Flat Panel Monitors, Net Sales et January.

```
DB('Budget', 'California', '15" Flat Panel Monitors', 'Net Sales', 'January')
```

Si la fonction est utilisée pour référencer des dimensions à plusieurs hiérarchies, vous devez spécifier la hiérarchie. Dans cet exemple, l'élément Category2 existe dans la hiérarchie ByCategory de la dimension ProductsCube.

```
DB('ProductsCube', 'ByCategory': 'Category2', ...)
```

## ISLEAF

ISLEAF renvoie 1 si une cellule spécifiée est une cellule feuille (identifiée seulement par des éléments feuille/simples). Si la cellule spécifiée est identifiée par des éléments consolidés, la fonction renvoie 0.

Cette fonction est valide uniquement dans les règles TM1.

La fonction ISLEAF ne peut être utilisée dans des processus TurboIntegrator. La présence de cette fonction dans un processus empêche sa compilation.

### Syntaxe

```
ISLEAF
```

### Arguments

Aucun.

## Exemple

Vous pouvez utiliser ISLEAF dans une instruction IF afin de tester si une cellule en cours est une cellule feuille. Par exemple :

```
[]=IF((ISLEAF=1),TrueStatement, FalseStatement);
```

Exécute l'instruction TrueStatement si la cellule en cours est une cellule feuille, sinon, elle exécute l'instruction FalseStatement.

## ISUNDEFINEDCELLVALUE

ISUNDEFINEDCELLVALUE compare la valeur passée avec la valeur du cube numérique par défaut, déterminée par la présence de la déclaration UNDEFVALS dans la règle de ce cube. La fonction renvoie 1 si la valeur passée est égale à la valeur par défaut du cube ; sinon, elle renvoie 0.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

ISUNDEFINEDCELLVALUE(*TestValue*, <Cube>)

### Arguments

Argument	Description
<i>TestValue</i>	Valeur numérique à comparer avec la valeur par défaut du cube.
<i>Cube</i>	Argument chaîne facultatif indiquant le cube dont la valeur par défaut doit être comparée.  Lorsque la fonction ISUNDEFINEDCELLVALUE est utilisée dans une règle, le cube est considéré comme cube objet si aucune autre valeur n'est spécifiée.  Lorsque cette fonction est utilisée dans un processus TI, un cube doit être spécifié.  Si le cube n'est pas spécifié, ou si une valeur non valide est spécifiée, la valeur 0 est utilisée pour la comparaison.

### Exemple

ISUNDEFINEDCELLVALUE(*TestValue*) renvoie 1 lorsque TestValue est la valeur non définie spéciale utilisée dans la règle d'un cube avec UNDEFVALS déclaré.

## UNDEF

UNDEF renvoie la valeur indéfinie. Cette fonction peut être utilisée pour empêcher le stockage de données dans un cube basé sur un test logique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
UNDEF
```

### Arguments

Aucun.

## Exemple

UNDEF renvoie la valeur indéfinie.

## UNDEFINEDCELLVALUE

UNDEFINEDCELLVALUE renvoie la valeur du cube numérique par défaut, déterminée par la présence de la déclaration UNDEFVALS dans la règle de ce cube.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

UNDEFINEDCELLVALUE (<Cube>)

## Arguments

Argument	Description
<i>Cube</i>	<p>Argument chaîne facultatif indiquant le cube dont la valeur par défaut doit être renvoyée.</p> <p>Lorsque la fonction UNDEFINEDCELLVALUE est utilisée dans une règle, le cube est considéré comme cube objet si aucune autre valeur n'est spécifiée.</p> <p>Lorsque cette fonction est utilisée dans un processus TI, un cube doit être spécifié.</p> <p>Si le cube n'est pas spécifié, ou si une valeur non valide est spécifiée, la valeur 0 est renvoyée.</p>

## Exemple

UNDEFINEDCELLVALUE renvoie 0 si elle est utilisée dans la règle d'un cube et que UNDEFVALS n'est pas déclarée, ou si elle est utilisée dans un processus TI.

UNDEFINEDCELLVALUE renvoie la valeur non définie spéciale lorsqu'elle est utilisée dans la règle d'un cube avec UNDEFVALS déclarée.

UNDEFINEDCELLVALUE ( ' Examp1eCube ' ) renvoie la valeur par défaut de 'ExampleCube', ou 0 si 'ExampleCube' n'existe pas.

## UNDEFVALS

L'insertion d'UNDEFVALS dans les règles d'un cube remplace la valeur par défaut utilisée dans le cube, zéro, par une valeur *indéfinie spéciale*. Comme d'autres fonctions de règles, la fonction UNDEFVALS s'applique uniquement au cube associé à la règle dans laquelle elle réside.

Il s'agit d'une fonction de règles TM1, valide uniquement dans les règles TM1.

L'utilisation d'UNDEFVALS a des conséquences sur le stockage des données dans le cube et leur extraction.

### • Stockage de données

Dans le cas d'un cube dont les règles ne contiennent pas la fonction UNDEFVALS, la valeur par défaut est zéro. En cas de tentative de stockage d'un zéro dans une cellule du cube, cette demande est ignorée puisqu'il s'agit de la valeur de stockage par défaut et que cette requête consommerait inutilement de l'espace mémoire. De la même façon, lorsqu'une cellule contient une valeur et que celle-ci est supprimée, rien n'est stocké dans la cellule.

Par contre, si la fonction UNDEFVALS est définie dans les règles du cube, la valeur par défaut devient alors une valeur *indéfinie spéciale*. Dans ce cas, lors d'une tentative de stockage d'un zéro dans une cellule du cube, cette valeur est réellement stockée, tout comme toutes autre valeur différente de zéro.

La valeur *indéfinie spéciale* n'est qu'une valeur d'exécution, renvoyée par les demandes pour les valeurs des cellules. Elle n'est jamais réellement stockée dans une cellule dans la mémoire, ni enregistrée sur le disque. L'insertion d'UNDEFVALS dans une règle de cube est sans effet sur l'utilisation de la mémoire ou sur le stockage sur disque, sauf pour les cellules qui contiennent réellement la valeur zéro. Lorsque la fonction UNDEFVALS est incluse dans la règle d'un cube, les valeurs zéro de ce cube consomment de l'espace mémoire et sont enregistrées sur le disque, comme n'importe quelle autre valeur. Si la règle ne contient pas UNDEFVALS, les cellules contenant un zéro ne sont pas stockées dans la mémoire, ni enregistrées sur le disque.

#### • Extraction de données

Dans le cas d'un cube où UNDEFVALS est absente des règles, la valeur par défaut est zéro. Lors de l'extraction d'une cellule alors qu'aucune valeur n'est actuellement stockée dans le cube, la valeur zéro (valeur par défaut) est renvoyée. Cela signifie que l'application ne peut pas savoir si une cellule existe et contient un zéro, ou si elle n'existe pas (comme dans le cas des données à faible densité).

Par contre, si UNDEFVALS est définie dans les règles du cube, la valeur par défaut devient alors une valeur *indéfinie spéciale*. Dans ce cas, lors de l'extraction d'une cellule non existante, la valeur extraite sera cette valeur *indéfinie spéciale*. Ceci permet de distinguer une cellule qui n'existe pas (renvoi d'une valeur *indéfinie spéciale*) d'une cellule existante mais dont la valeur est zéro (renvoi de zéro). Les clients conçus pour fonctionner avec TM1, et qui sont susceptibles de faire appel à un cube contenant la fonction UNDEFVALS, doivent être prêts à gérer une valeur de cellule *indéfinie spéciale* à la place d'un zéro. La fonction d'API `TM1ValIsUndefined` permet au client de détecter si une valeur renvoyée par TM1 est cette valeur *indéfinie spéciale*. Pour plus d'informations sur la fonction d'API `TM1ValIsUndefined`, reportez-vous au document *TM1 API*.

**Remarque :** Cette valeur *indéfinie spéciale* n'est pas la valeur renvoyée par la fonction `UNDEF()` de TurboIntegrator. La valeur renvoyée par `UNDEF()` correspond à la valeur indéfinie utilisée pour des opérations, comme une tentative de division par zéro ou d'obtention du logarithme d'un nombre non admis, etc.

Dans TurboIntegrator, pour les opérations arithmétiques normales (+, -, \*, /, \, ^) et les comparaisons arithmétiques normales (<, >, >=, <=, =, <>), la valeur *indéfinie spéciale* est considérée comme un zéro. Pour cette raison, le code suivant ne fonctionne pas :

```
NoCellVal = UndefinedCellValue( 'cube-name' );
If ( vv = NoCellVal );
```

Dans cette comparaison, `NoCellVal`, qui est la valeur *indéfinie spéciale* d'un cube UNDEFVALS, est traité comme un zéro. Cela signifie que la comparaison est réellement `If ( vv = 0 )`.

Dans TurboIntegrator, vous devez utiliser `IsUndefinedCellValue` pour tester si la valeur d'une cellule est la valeur *indéfinie spéciale*. Par exemple :

```
vv = CellGetN( 'cube-name', elements-list);
if ( IsUndefinedCellValue( vv, 'cube-name' ) = 1 );
#the cells does not exist
cell_does_not_exist = 1;
else;
#cell exists
cell_does_not_exist = 0;
Endif;
```

## Syntaxe

UNDEFVALS

## Arguments

Aucun.



## Fonctions de règles de date et heure

Fonctions de règles qui utilisent des dates et des heures.

### DATE

DATE renvoie la chaîne de date au format *aa-mm-jj* ou *aaaa-mm-jj* pour un numéro de série donné.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
DATE(SerialNumber, ReturnFourDigitYear)
```

Argument	Description
<i>SerialNumber</i>	Date exprimée sous forme de séquence de chiffres.
<i>ReturnFourDigitYear</i>	<p>Argument booléen facultatif qui détermine si la fonction DATE renvoie une chaîne qui utilise la notation à deux chiffres ou la notation à quatre chiffres pour l'année.</p> <p>Si <i>ReturnFourDigitYear</i> est vrai, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 9999 en utilisant la notation à quatre chiffres pour l'année. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 2936549 à 31 déc. 9999.</p> <p>Si <i>ReturnFourDigitYear</i>, ou si cet argument facultatif est omis dans la fonction DATE, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 2059 en utilisant la notation à deux chiffres pour l'année. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 36524 à 31 déc. 2059.</p> <p>Si <i>ReturnFourDigitYear</i> est faux ou omis et si vous spécifiez un numéro de date supérieur à 36524, le numéro de date utilisé par la fonction est déterminé par la formule <math>n - 36525</math>. Par exemple, si vous spécifiez comme numéro de date 36530, alors <math>36530 - 36525 = 5</math>. Dans ce cas, DATE utilise 5 comme numéro de date et renvoie la date 6 jan. 1960.</p>

#### Exemple

DATE(13947) renvoie 98-03-09.

DATE(13947, 1) renvoie 1998-03-09.

### DATES

DATES renvoie une chaîne de date dans le format '*aa-mm-jj*' ou '*aaaa-mm-jj*', correspondant à une année, un mois et un jour donnés.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
DATES(year, month, day)
```

Argument	Description
year	Année exprimée dans le format aa ou aaaa.
month	Mois exprimé dans le format mm.
day	Jour exprimé dans le format jj.

### Exemple

DATES(98, 2, 10) renvoie '98-02-10'.

DATES(1998, 2, 10) renvoie '1998-02-10'.

## DAY

DAY renvoie une valeur numérique pour le jour d'une chaîne de date donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
DAY(DateString)
```

Argument	Description
DateString	Chaîne de date exprimée dans le format AA-MM-JJ ou AAAAA-MM-JJ.

### Exemple

DAY('02-05-25') renvoie 25.

## DAYNO

DAYNO renvoie une séquence de chiffres correspondant à une date donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

**Remarque :** DAYNO peut renvoyer des numéros de date pour les chaînes de date commençant au 1er janvier 1960 (chaîne de date 1960-01-01 ou 60-01-01). Pour les dates au-delà du 31 décembre 2059, indiquez les quatre chiffres de l'année dans la chaîne de date. Par exemple, la chaîne de date pour le 5 janvier 2061 sera 2061-01-05.

### Syntaxe

```
DAYNO('DateString')
```

Argument	Description
DateString	Chaîne de date exprimée dans le format AA-MM-JJ ou AAAAA-MM-JJ.

### Exemple

DAYNO('98-03-09') renvoie 13947.

## MONTH

MONTH renvoie une valeur numérique pour le mois d'une chaîne de date donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
MONTH(date)
```

Argument	Description
date	Chaîne de date exprimée dans le format AA-MM-JJ ou AAAA-MM-JJ.

### Exemple

MONTH('02-05-25') renvoie 5.

## NOW

NOW renvoie la date et l'heure en cours sous forme d'une série de chiffres.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
NOW
```

### Arguments

Aucun.

### Exemple

NOW renvoie la date et l'heure en cours sous forme d'une série de chiffres.

## TIME

TIME renvoie une chaîne, au format HH:MM, représentant l'heure système sur le serveur TM1.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
TIME
```

### Arguments

Aucun.

### Exemple

Si l'heure du système est 09:33:00 AM, TIME renvoie la chaîne '09:33:00'.

Si l'heure du système est 9:33 PM, TIME renvoie la chaîne '21:33'.

## TIMST

TIMST renvoie une chaîne date/heure formatée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

TIMST(datetime, format, ExtendedYears)

Argument	Modificateur/ Description
datetime	Série de numéros représentant une date/heure. La partie entière du nombre spécifie la date et la partie décimale spécifie l'heure du jour. Le jour numéro 0 correspond à '60-01-01'. Les nombres négatifs correspondent aux années antérieures. Les années du 21ème siècle, jusqu'à 2059, sont représentées par les années 00 à 59. Une heure est 1/24ème de jour, une minute 1/60ème d'heure et une seconde 1/60ème de minute.
format	Chaîne formatant le résultat de la fonction. Tous les arguments se trouvant dans l'argument de format apparaissent dans le résultat, à l'exception des caractères suivants, qui renvoient les valeurs de composant date/heure :
	\y les deux derniers chiffres de l'année (97, 98, etc.).
	\Y les quatre chiffres de l'année (1997, 1998, etc.).
	\m les deux chiffres du mois (01 à 12).
	\M l'abréviation du mois (JAN, FEV, etc.).
	\d les deux chiffres du jour (01 à 31).
	\D le chiffre du jour (1 à 31).
	\h l'heure en format de 24 heures (00 à 23).
	\H l'heure en format de 12 heures (1 à 12).
	\i la minute (00 à 59).

Argument	Modificateur/ Description
	\s la seconde (00 à 59).
	\p a.m. ou p.m. (avant ou après midi)
ExtendedYears	<p>Ce paramètre booléen facultatif spécifie si la fonction renvoie une date se trouvant dans la plage 1960-2059 ou dans la plage 1960-9999.</p> <p>Si ExtendedYears est vrai, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 9999. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 2936549 à 31 déc. 9999.</p> <p>Si ExtendedYears est faux, ou si cet argument facultatif est omis dans la fonction TIMST, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 2059. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 36524 à 31 déc. 2059.</p> <p>Si ExtendedYears est faux ou omis et si vous spécifiez un numéro de date supérieur à 36524, le numéro de date utilisé par la fonction est déterminé par la formule <math>n - 36525</math>. Par exemple, si vous spécifiez comme numéro de date 36530, alors <math>36530 - 36525 = 5</math>. Dans ce cas, TIMST utilise 5 comme numéro de date et renvoie la date 6 jan. 1960.</p>

### Exemple

TIMST(366.0000, '\D \M \Y') renvoie '1 JAN 1961'.

TIMST(366.5000, '\H\p \imin\ssec') renvoie '12p.m. 00min00sec'.

TIMST(366.1000, 'Le \D \M \Y à \H\p \imin\ssec') renvoie 'Le 1 JAN 1 1961 à 2a.m. 24min00sec'.

TIMST(11111.1100, 'On \M \D, \Y at \H\p \imin\ssec') returns 'On JUN 3,1990 at 2a.m. 38min24sec'.

### TIMVL

TIMVL renvoie la valeur numérique d'un composant (année, mois, etc.) d'une valeur date-heure.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
TIMVL(datetime, type, ExtendedYears)
```

Argument	Modificateur et Description
datetime	<p>Date et heure exprimées sous forme de séquence de chiffres.</p> <p>La partie entière du nombre spécifie la date et la partie décimale spécifie l'heure du jour. Le jour numéro 0 correspond à '60-01-01'. Les nombres négatifs correspondent aux années antérieures. Les années du 21ème siècle, jusqu'à 2059, sont représentées par les années 00 à 59. Une heure est 1/24ème de jour, une minute 1/60ème d'heure et une seconde 1/60ème de minute.</p>
type	Caractère spécifiant le type de composant à extraire. Les types d'argument ci-dessous sont valides :
	Y valeur d'année (1997, 1998, etc.)
	M valeur de mois (1 à 12)
	D valeur de jour (1 à 31)
	H valeur d'heure (0 à 23)
	I valeur de minute (00 à 59)
	S valeur de seconde (00 à 59)

Argument	Modificateur et Description
ExtendedYears	<p>Ce paramètre booléen facultatif spécifie si la fonction renvoie une date se trouvant dans la plage 1960-2059 ou dans la plage 1960-9999.</p> <p>Si ExtendedYears est vrai, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 9999. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 2936549 à 31 déc. 9999.</p> <p>Si ExtendedYears est faux, ou si cet argument facultatif est omis dans la fonction TIMVL, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 2059. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 36524 à 31 déc. 2059.</p> <p>Si ExtendedYears est faux ou omis et si vous spécifiez un numéro de date supérieur à 36524, le numéro de date utilisé par la fonction est déterminé par la formule <math>n - 36525</math>. Par exemple, si vous spécifiez comme numéro de date 36530, alors <math>36530 - 36525 = 5</math>. Dans ce cas, TIMVL utilise 5 comme numéro de date et renvoie la date 6 jan. 1960.</p>

### Exemple

TIMVL(111111.1100, 'Y') renvoie 1990.

TIMVL(111111.1100, 'H') renvoie 2.

### TODAY

TODAY renvoie la date en cours dans le format aa-mm-jj.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
TODAY(<ReturnFourDigitYear>)
```

Argument	Description
ReturnFourDigitYear	<p>Argument booléen facultatif qui détermine si la fonction TODAY renvoie une chaîne qui utilise la notation à deux chiffres ou la notation à quatre chiffres pour l'année.</p> <p>Si ReturnFourDigitYear est vrai, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 9999 en utilisant la notation à quatre chiffres pour l'année. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 2936549 à 31 déc. 9999.</p> <p>Si ReturnFourDigitYear est faux, ou si cet argument facultatif est omis dans la fonction TODAY, la fonction renvoie une date qui se trouve dans la plage 1 jan. 1960 à 31 déc. 2059 en utilisant la notation à deux chiffres pour l'année. Le numéro de date 0 correspond à 1 jan. 1960 et le numéro 36524 à 31 déc. 2059.</p> <p>Si ReturnFourDigitYear est faux ou omis et si vous spécifiez un numéro de date supérieur à 36524, le numéro de date utilisé par la fonction est déterminé par la formule <math>n - 36525</math>. Par exemple, si vous spécifiez comme numéro de date 36530, alors <math>36530 - 36525 = 5</math>. Dans ce cas, TODAY utilise 5 comme numéro de date et renvoie la date 6 jan. 1960.</p>

### Exemple

P1=TODAY(1) renvoie une chaîne de données au format AAAA-MM-JJ comme 2009-06-05.

P1=TODAY(0) renvoie une chaîne de données au format AA-MM-JJ comme 09-06-05.

## YEAR

YEAR renvoie une valeur numérique pour l'année dans une chaîne de date donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

YEAR(date)

Argument	Description
date	Chaîne de date dans le format AA-MM-JJ.

### Exemple

YEAR('02-05-25') renvoie 2.

## Fonctions de règles d'informations de dimensions

Fonctions de règles qui gèrent des informations de dimensions.



## DIMIX

DIMIX renvoie le numéro d'index d'un élément dans une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
DIMIX(server_name:dimension, element)
```

Argument	Description
dimension	Nom de dimension valide qualifié à l'aide du nom du serveur.
element	Nom d'un élément de la dimension. Si l'élément n'appartient pas à la dimension spécifiée, la fonction renvoie 0.

### Exemple

Brazil a la valeur d'index 3 dans la dimension Region. L'exemple renvoie 3.

```
DIMIX('planning_sample:Region', 'Brazil')
```

## DIMNM

DIMNM renvoie l'élément d'une dimension correspondant à l'argument index.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
DIMNM(server_name:dimension, index)
```

Argument	Description
dimension	Nom de dimension valide qualifié à l'aide du nom du serveur.
index	Valeur inférieure ou égale au nombre d'éléments dans la dimension. Si l'argument est inférieur à 1 ou supérieur au nombre d'éléments de la dimension, la fonction renvoie 0.

### Exemple

Cet exemple renvoie 'Belgique', qui est l'élément dans la dimension Région avec la valeur d'index 2.

```
DIMNM(planning_sample:'Region', 2)
```

## DIMSIZ

DIMSIZ renvoie le nombre d'éléments d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
DIMSIZ(dimension)
```

Argument	Description
dimension	Nom de dimension valide. Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .

## Exemple

Si la dimension Comptes contient 19 éléments, l'exemple renvoie la valeur 19.

```
DIMSIZ('Comptes')
```

## DNEXT

DNEXT renvoie le nom de l'élément qui suit celui spécifié comme argument de la fonction.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
DNEXT(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide. Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

## Exemple

Si la dimension Site contient, dans cet ordre, les éléments California, Oregon et Washington, l'exemple renvoie Washington.

```
DNEXT("Site", "Oregon")
```

## DNLEV

DNLEV renvoie le nombre de niveaux d'une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

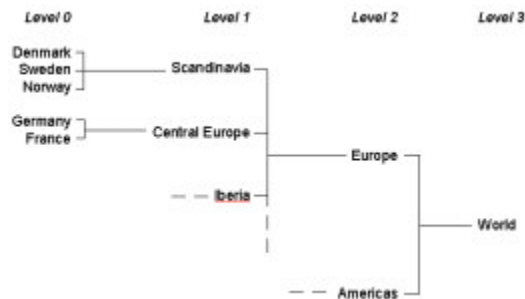
```
DNLEV(dimension)
```

Argument	Description
dimension	Nom de dimension valide. Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .

### Exemple

```
DNLEV('Region')
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3).



Il y a quatre niveaux dans la dimension Region, et, par conséquent, l'exemple renvoie la valeur 4.

### DTYPE

DTYPE renvoie des informations sur le type d'élément de l'élément spécifié. DTYPE renvoie N si l'élément est numérique, S s'il s'agit d'une chaîne et C si l'élément est consolidé.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
DTYPE(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension.

### Exemple

L'élément Europe est un élément consolidé de la dimension Region et l'exemple renvoie donc C.

```
DTYPE('Region', 'Europe')
```

### TABDIM

TABDIM renvoie le nom de dimension correspondant à l'argument d'index.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
TABDIM(cube, index)
```

Argument	Description
cube	Nom de cube valide.
index	Valeur positive inférieure ou égale au nombre total de dimensions du cube.

## Exemple

Le cube CubeVentes compte cinq dimensions : compte1, réelebud, modèle, mois et région. L'exemple renvoie modèle, qui est la troisième dimension de CubeVentes.

```
TABDIM('CubeVentes',3)
```

## Fonctions de règles d'informations d'éléments

Fonctions de règles qui gèrent des informations d'éléments.

### ELCOMP

ELCOMP renvoie le nom d'un enfant d'un élément consolidé d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ELCOMP(dimension, element, position)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément consolidé dans la dimension.
position	Valeur positive inférieure ou égale au nombre total d'enfants dans l'élément spécifié.

## Exemple

Dans la dimension Region, l'élément consolidé Central Europe est une consolidation des enfants Germany et France. Germany est en seconde position dans cette consolidation. Par conséquent, l'exemple renvoie Germany.

```
ELCOMP('Region','Central Europe',2)
```

### ELCOMPN

ELCOMPN renvoie le nombre de composants d'un élément spécifié. Si l'argument de l'élément n'est pas un élément consolidé, la fonction renvoie la valeur 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ELCOMPN(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément consolidé dans la dimension.

## Exemple

Dans la dimension Region, l'élément Scandinavia est une consolidation de trois éléments. L'exemple renvoie 3.

```
ELCOMPN('Region', 'Scandinavia')
```

## ElementComponent

ElementComponent renvoie le nom d'un enfant d'un élément consolidé d'une dimension spécifiée. Si l'argument de l'élément n'est pas un élément consolidé, la fonction renvoie la valeur 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementComponent(dimension, hierarchy, element, position)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément consolidé dans la dimension.
position	Valeur positive inférieure ou égale au nombre total d'enfants dans l'élément spécifié.

## Exemple

Dans la dimension Region, l'élément consolidé Central Europe est une consolidation des enfants Germany et France. Germany est en seconde position dans cette consolidation. Par conséquent, l'exemple renvoie Germany.

```
ElementComponent('Region', 'Europe', 'Central Europe', 2)
```

## ElementComponentCount

ElementComponentCount renvoie le nombre de composants d'un élément spécifié. Si l'argument de l'élément n'est pas un élément consolidé, la fonction renvoie la valeur 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementComponentCount(dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément consolidé dans la dimension.

### Exemple

Dans la dimension Region, l'élément Scandinavia est une consolidation de trois éléments. L'exemple renvoie 3.

```
ElementComponentCount('Region', '', 'Scandinavia')
```

## ElementCount

ElementCount renvoie le nombre d'éléments d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementCount(dimension, hierarchy)
```

Argument	Description
dimension	Nom de dimension valide. Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .
hierarchy	Nom de la hiérarchie dans la dimension.

### Exemple

Si la hiérarchie Receivables de la dimension Accounts contient 19 éléments, l'exemple renvoie la valeur 19.

```
ElementCount('Accounts', 'Receivables')
```

## ElementFirst

ElementFirst renvoie le premier élément d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementFirst(server_name:dimension, hierarchy)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.

## Exemple

Si la hiérarchie North America de la dimension Location contient les éléments California, Oregon et Washington, dans cet ordre, l'exemple renvoie California.

```
ElementFirst("planning_sample:Location", "North America")
```

## ElementIndex

ElementIndex renvoie le numéro d'index d'un élément dans une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementIndex(server_name:dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide qualifié à l'aide du nom du serveur.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension. Si l'élément n'appartient pas à la dimension spécifiée, la fonction renvoie 0.

## Exemple

Brazil a la valeur d'index 3 dans la dimension Region. L'exemple renvoie 3.

```
ElementIndex('planning_sample:Region', 'South America', 'Brazil')
```

## ElementIsAncestor

ElementIsAncestor détermine si element1 est un ancêtre d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un ancêtre d'element2, sinon elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementIsAncestor(dimension, hierarchy, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

## Exemple

Dans la hiérarchie Western de la dimension Region, l'élément Europe est un ancêtre de Germany. L'exemple renvoie 1.

```
ElementIsAncestor('Region', 'Western', 'Europe', 'Germany')
```

## ElementIsComponent

ElementIsComponent détermine si element1 est un enfant d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un enfant d'element2, sinon elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementIsComponent(dimension, hierarchy, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

## Exemple

Dans la dimension Region, l'élément Central Europe est une consolidation des deux éléments Germany et France. L'exemple renvoie 1.

**Remarque :** Notez que cette fonction ne renvoie 1 que pour les enfants immédiats. Dans l'exemple ci-dessus, Germany est un enfant de Central Europe. De plus, Central Europe est un enfant de Europe.

```
ElementIsComponent('Region', 'Countries', 'Germany', 'Central Europe')
```

Cependant, puisque la fonction ne renvoie 1 que pour les enfants immédiats, l'exemple ci-dessous renvoie 0.

```
ElementIsComponent('Region', 'Countries', 'Germany', 'Europe')
```

## ElementIsParent

ElementIsParent détermine si element1 est un parent d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un parent d'element2 ; sinon elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementIsParent(dimension, hierarchy, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.



Argument	Description
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

### Exemple

Dans la dimension Region, l'élément consolidé Central Europe est le parent de Germany et France. Par conséquent, l'exemple renvoie 1.

**Remarque :** Notez que cette fonction ne renvoie 1 que pour les parents immédiats. Dans l'exemple ci-dessus, Europe est un parent de Central Europe. De plus, Central Europe est un parent de Germany.

```
ElementIsParent('Region', 'Countries', 'Central Europe', 'Germany')
```

Toutefois, Europe n'étant pas un parent immédiat de Germany, l'exemple ci-dessous renvoie 0.

```
ElementIsParent('Region', 'Countries', 'Europe', 'Germany')
```

## ElementLevel

ElementLevel renvoie le niveau d'un élément dans une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

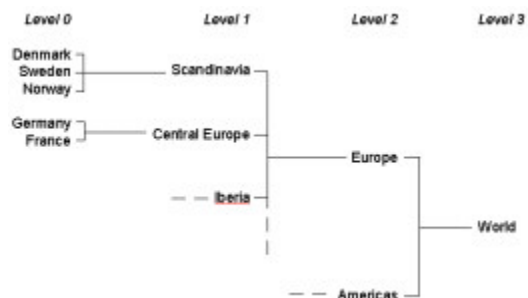
```
ElementLevel(dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension.

### Exemple

```
ElementLevel('Region', 'Countries', 'Europe')
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3). L'exemple renvoie 2, Europe étant un élément du niveau 2.



## ElementName

ElementName renvoie l'élément d'une dimension correspondant à l'argument index.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementName(server_name:dimension, hierarchy, index)
```

Argument	Description
dimension	Nom de dimension valide qualifié à l'aide du nom du serveur.
hierarchy	Nom de la hiérarchie dans la dimension.
index	Valeur inférieure ou égale au nombre d'éléments dans la dimension.  Si l'argument est inférieur à 1 ou supérieur au nombre d'éléments de la dimension, la fonction renvoie 0.

### Exemple

Cet exemple renvoie 'Belgium', l'élément dans la hiérarchie Countries de la dimension Region dont la valeur d'index est 2.

```
ElementName(planning_sample:'Region', 'Countries', 2)
```

## ElementNext

ElementNext renvoie le nom de l'élément qui suit celui spécifié comme argument de la fonction.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementNext(dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide.  Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

## Exemple

Si la dimension Site contient, dans cet ordre, les éléments California, Oregon et Washington, l'exemple renvoie Washington.

```
ElementNext("Location","Cities", "Oregon")
```

## ElementParent

ElementParent renvoie le parent d'un élément d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementParent(dimension, hierarchy, element, index)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension.
index	Valeur positive inférieure ou égale au nombre total d'éléments consolidés (parents) utilisant l'argument d'élément comme enfant.

## Exemple

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. Dans la structure de la dimension Model, Total Wagons est défini en premier et Total 4WD est défini en second.

```
ElementParent('Model', 'Automobile', 'Wagon 4WD', 2)
```

L'exemple renvoie Total 4WD, deuxième instance d'un parent de Wagon 4WD dans la dimension Model.

## ElementParentCount

ElementParentCount renvoie le nombre de parents d'un élément d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ElementParentCount(dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension.

## Exemple

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. La fonction renvoie 2.

```
ElementParentCount('Model', 'Automobile', 'Wagon 4WD')
```

## ElementType

ElementType renvoie des informations sur le type d'élément de l'élément spécifié. ElementType renvoie N si l'élément est un élément numérique, S (pour String) si c'est une chaîne et C si l'élément est un élément consolidé.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementType(dimension, hierarchy, element)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element	Nom d'un élément de la dimension.

## Exemple

L'élément Europe est un élément consolidé de la dimension Region et l'exemple renvoie donc C.

```
ElementType('Region', 'Countries', 'Europe')
```

## ElementWeight

ElementWeight renvoie la pondération d'un enfant d'un élément consolidé.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

## Syntaxe

```
ElementWeight(dimension, hierarchy, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
hierarchy	Nom de la hiérarchie dans la dimension.
element1	Nom d'un élément consolidé dans la dimension.
element2	Nom d'un enfant de l'élément consolidé.

### Exemple

L'élément Coûts variables, qui est un enfant de Marge brute, a une pondération de -1. L'exemple suivant renvoie -1 :

```
ElementWeight('Account1', 'SubAccount1', 'Gross margin', 'Variable Costs')
```

## ELISANC

ELISANC détermine si element1 est un ancêtre d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un ancêtre d'element2, sinon elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELISANC(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

### Exemple

Dans la dimension Region, l'élément Europe est un ancêtre de Allemagne. L'exemple renvoie 1.

```
ELISANC('Région', 'Europe', 'Allemagne')
```

## ELISCOMP

ELISCOMP détermine si element1 est un enfant d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un enfant d'element2 ; sinon, elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELISCOMP(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

### Exemple

Dans la dimension Region, l'élément Central Europe est une consolidation des deux éléments, Germany et France. L'exemple suivant renvoie 1.

**Remarque :** Notez que cette fonction ne renvoie 1 que pour les enfants immédiats. Dans cet exemple, Germany est un enfant de Central Europe. De plus, Central Europe est un enfant de Europe.

```
ELISCOMP('Region', 'Germany', 'Central Europe')
```

Cependant, puisque la fonction ne renvoie 1 que pour les enfants immédiats, l'exemple ci-dessous renvoie 0.

```
ELISCOMP('Region', 'Germany', 'Europe')
```

## ELISPAR

ELISPAR détermine si element1 est un parent d'element2 dans la dimension spécifiée. La fonction renvoie 1 si element1 est un parent d'element2 ; sinon elle renvoie 0.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELISPAR(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément de la dimension.
element2	Nom d'un élément de la dimension.

### Exemple

Dans la dimension Region, l'élément consolidé Central Europe est le parent de Germany et France. Par conséquent, l'exemple suivant renvoie 1.

**Remarque :** Notez que cette fonction ne renvoie 1 que pour les parents immédiats. Dans cet exemple, Europe est un parent de Central Europe. De plus, Central Europe est un parent de Germany.

```
ELISPAR('Region', 'Central Europe', 'Germany')
```

Toutefois, Europe n'étant pas un parent immédiat de Germany, l'exemple ci-dessous renvoie 0.

```
ELISPAR('Region', 'Europe', 'Germany')
```

## ELLEV

ELLEV renvoie le niveau d'un élément d'une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

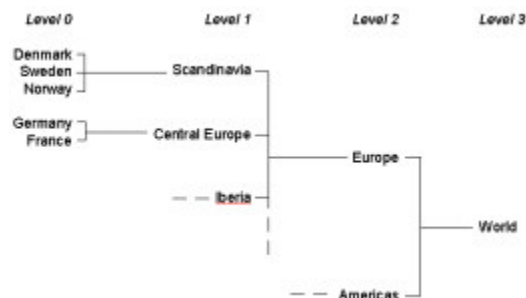
```
ELLEV(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension.

## Exemple

```
ELLEV('Region', 'Europe')
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3). L'exemple renvoie 2, Europe étant un élément du niveau 2.



## ELPAR

ELPAR renvoie le parent d'un élément d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELPAR(dimension, element, index)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension.
index	Valeur positive inférieure ou égale au nombre total d'éléments consolidés (parents) utilisant l'argument d'élément comme enfant.

## Exemple

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. Dans la structure de la dimension Model, Total Wagons est défini en premier et Total 4WD est défini en second.

```
ELPAR('Model', 'Wagon 4WD', 2)
```

L'exemple renvoie Total 4WD, deuxième instance d'un parent de Wagon 4WD dans la dimension Model.

## ELPARN

ELPARN renvoie le nombre de parents d'un élément d'une dimension spécifiée.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELPARN(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension.

### Exemple

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. La fonction renvoie 2.

```
ELPARN('Model', 'Wagon 4WD')
```

## ELWEIGHT

ELWEIGHT renvoie la pondération d'un enfant d'un élément consolidé.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
ELWEIGHT(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément consolidé dans la dimension.
element2	Nom d'un enfant de l'élément consolidé.

### Exemple

L'élément Coûts variables, qui est un enfant de Marge brute, a une pondération de -1.

L'exemple suivant renvoie -1 :

```
ELWEIGHT('Compte1', 'Marge brute', 'Coûts variables')
```

## LevelCount

LevelCount renvoie le nombre de niveaux dans une dimension.

Cette fonction est valide dans les règles TM1 et les processus TurboIntegrator.

### Syntaxe

```
LevelCount(dimension, hierarchy)
```

Argument	Description
dimension	Nom de dimension valide.  Certaines installations peuvent devoir qualifier le nom de la dimension avec le nom du serveur, comme dans <code>server_name:dimension</code> .

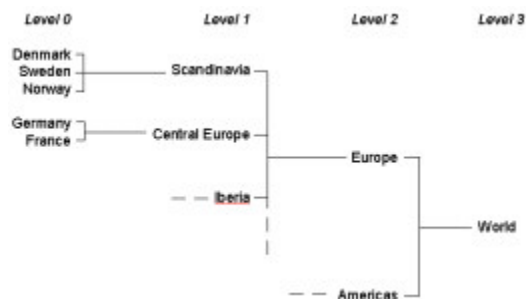


Argument	Description
hierarchy	Nom de la hiérarchie dans la dimension.

### Exemple

```
LevelCount('Region', 'Countries')
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3).



Il y a quatre niveaux dans la dimension Region, et, par conséquent, l'exemple renvoie la valeur 4.

## Fonctions de règles financières

Fonctions de règles utilisées pour gérer des informations financières.

### VC

FV renvoie la valeur d'une annuité au moment du dernier versement. Une annuité est une série de versements faits à un intervalle régulier. On suppose que les versements sont faits à la fin de chaque période.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
FV(payment, interest, periods)
```

Argument	Description
payment	Montant de chaque versement.
interest	Taux d'intérêt payé pour chaque période.
periods	Nombre de périodes dans l'annuité.

### Exemple

Cet exemple renvoie la valeur d'une annuité après 5 ans, un versement de \$1 000 dollars étant fait annuellement à un taux intérêt de 14%.

```
FV(1000, .14, 5)
```

## PAYMT

PAYMT renvoie la somme payée par une annuité à partir de sa valeur initiale (le principal), d'un taux d'intérêt et d'un nombre de périodes. Une annuité est une série de versements faits à un intervalle régulier. Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
PAYMT(principal, interest, periods)
```

Argument	Description
principal	Valeur présente, ou valeur totale en cours d'une série de versements futurs.
interest	Taux d'intérêt payé pour chaque période.
periods	Nombre de périodes dans l'annuité. On suppose que les versements sont faits à la fin de chaque période.

### Exemple

Cet exemple renvoie le paiement pour une annuité de 5 ans payée annuellement, le principal étant \$100 000 et le taux d'intérêt 14%.

```
PAYMT(100000, 14, 5)
```

## PV

PV renvoie la valeur initiale d'une annuité (son principal).

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
PV(payment, interest, periods)
```

Argument	Description
payment	Montant de chaque versement.
interest	Taux d'intérêt payé pour chaque période.
periods	Nombre de périodes dans l'annuité. On suppose que les versements sont faits à la fin de chaque période.

### Exemple

Cet exemple renvoie la valeur du principal d'une annuité pour laquelle seront faits 5 paiements annuels de \$1 000 à un taux d'intérêt de 14%.

```
PV(1000, .14, 5)
```

## Fonctions de règles de hiérarchie

Fonctions permettant de gérer des hiérarchies dans des règles.

### Hierarchy

S'il n'y a qu'une seule hiérarchie incluse dans la dimension fournie, la fonction Hierarchy renvoie le nom de la hiérarchie. Sinon, elle renvoie une chaîne vide. Hierarchy est valide uniquement dans des règles TM1.

Avec l'ajout de la prise en charge de plusieurs hiérarchies dans TM1, il est nécessaire d'identifier les hiérarchies qui sont en contexte lorsque plusieurs hiérarchies sont utilisées.

La fonction Hierarchy ne doit pas être utilisée dans des processus TurboIntegrator. La présence de cette fonction dans un processus empêche sa compilation.

#### Syntaxe

```
Hierarchy (DimName);
```

Argument	Description
DimName	Nom de dimension valide.

#### Exemple

Cet exemple renvoie 'Quarter', qui est la seule hiérarchie de la dimension Quarter.

```
Hierarchy ('Quarter');
```

### HierarchyCount

HierarchyCount renvoie le nombre de hiérarchies dans la dimension fournie. HierarchyCount est valide uniquement dans des règles TM1.

La fonction HierarchyCount ne doit pas être utilisée dans des processus TurboIntegrator. La présence de cette fonction dans un processus empêche sa compilation.

#### Syntaxe

```
HierarchyCount (DimName);
```

Argument	Description
DimName	Nom de dimension valide.

#### Exemple

Cet exemple renvoie 3, c'est-à-dire le nombre de hiérarchies dans la dimension 'model'.

```
HierarchyCount ('model');
```

### HierarchyIndex

HierarchyIndex renvoie un index basé sur 1 si la hiérarchie se trouve dans la dimension fournie. sinon, 0 est renvoyé. HierarchyIndex est valide uniquement dans des règles TM1.

La fonction HierarchyIndex ne doit pas être utilisée dans des processus TurboIntegrator. La présence de cette fonction dans un processus empêche sa compilation.

## Syntaxe

```
HierarchyIndex (DimName, HierName);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de hiérarchie valide dont vous souhaitez connaître la position d'index dans <i>DimName</i> .

## Exemple

Cet exemple renvoie 3, c'est-à-dire la position d'index de la hiérarchie CustomerTarget dans la dimension 'model'.

```
HierarchyIndex ('model', 'CustomerTarget');
```

## HierarchyN

HierarchyN renvoie le nom de la hiérarchie à une position spécifiée dans la dimension fournie ou une chaîne vide si l'index est en dehors de la portée. HierarchyN est valide uniquement dans des règles TM1.

La fonction HierarchyN ne doit pas être utilisée dans les processus TurboIntegrator. La présence de cette fonction dans un processus empêche sa compilation.

## Syntaxe

```
HierarchyN (DimName, index);
```

Argument	Description
DimName	Nom de dimension valide.
index	Valeur inférieure ou égale au nombre de hiérarchies dans la dimension.  Si cet argument est inférieur à 1 ou supérieur au nombre de hiérarchies dans la dimension, la fonction renvoie 0.

## Exemple

Cet exemple renvoie 'CustomerTarget', qui est la troisième hiérarchie de la dimension 'model'.

```
HierarchyN ('model', 3);
```

## Fonctions de règles logiques

Opérateurs logiques à utiliser dans les règles.

## CONTINUE

Lorsque cette fonction est incluse comme composant d'une expression de règles, elle permet l'exécution d'une règle suivante avec la même définition de zone. Normalement, TM1 exécute uniquement la première règle rencontrée pour une zone donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

```
CONTINUE
```

## Arguments

Aucun.

## Exemple

```
['Jan']= if(!region @= 'Argentina',10,CONTINUE);
```

```
['Jan']=20;
```

Dans cet exemple, toutes les cellules identifiées par January et Argentina reçoivent la valeur 10. Les cellules identifiées par Jan et tout autre élément Region reçoivent la valeur 20.

## IF

IF renvoie une valeur si une expression logique que vous spécifiez est vraie (TRUE) et une autre valeur si elle est fausse (FALSE).

Cette fonction est valide uniquement dans les règles TM1.

TurboIntegrator utilise sa propre fonction IF qui est capable d'évaluer plusieurs expressions logiques.

## Syntaxe

```
IF(expression, true_value, false_value)
```

Argument	Description
expression	Valeur ou expression pouvant être évaluée comme vraie (TRUE) ou fausse (FALSE).
true_value	Valeur renvoyée si l'expression est vraie (TRUE).
false_value	Valeur renvoyée si l'expression est fausse (FALSE).

## Exemple

IF(1<2, 4, 5) renvoie 4.

IF(1>2, 'ABC', 'DEF') renvoie 'DEF'.

## STET

La fonction STET annule l'effet d'une règle sur un élément donné.

Il s'agit d'une fonction de règles TM1, valide uniquement dans les règles TM1. Non disponible dans des processus TurboIntegrator.

## Syntaxe

```
STET
```

## Arguments

Aucun.

## Exemple

Dans cet exemple, la règle spécifie que la valeur de Sales est toujours 100, sauf à l'intersection de Sales et de l'élément France de la dimension Region.

```
['Sales'] = IF(!Region @= 'France',STET, 100);
```

## Fonctions de règles mathématiques

Opérateurs mathématiques à utiliser dans les règles.

### ABS

ABS renvoie la valeur absolue d'un nombre.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
ABS(x)
```

Argument	Description
x	Nombre dont vous voulez trouver la valeur absolue.

#### Exemple

ABS(-1,2) renvoie 1,2

### ACOS

ACOS renvoie en radians l'angle dont le cosinus est x.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
ACOS(x)
```

Argument	Description
x	Cosinus de l'angle recherché. x doit être entre -1 et 1, sinon la fonction renvoie une erreur.

#### Exemple

ACOS(0) renvoie 1,5708.

### ASIN

ASIN renvoie, en radians, l'angle dont le sinus est x.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

#### Syntaxe

```
ASIN(x)
```

Argument	Description
x	Sinus de l'angle recherché. x doit être entre -1 et 1, sinon la fonction renvoie une erreur.

### Exemple

ASIN(1) renvoie 1,5708.

## ATAN

ATAN renvoie en radians l'angle dont la tangente est x. Le résultat est entre  $-\pi/2$  et  $+\pi/2$ .

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

ATAN(x)

Argument	Description
x	Tangente de l'angle recherché.

### Exemple

ATAN(1) renvoie 0,7854.

## COS

COS renvoie le cosinus d'un angle exprimé en radians.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

COS(x)

Argument	Description
x	Angle, exprimé en radians, dont vous recherchez le cosinus.

### Exemple

COS(0) renvoie 1.

## EXP

EXP renvoie l'antilogarithme naturel d'un nombre.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

EXP(x)

Argument	Description
x	Nombre dont vous voulez trouver l'antilogarithme naturel.

### Exemple

EXP(1) renvoie 2,71828.

## ENT

INT renvoie le plus grand entier inférieur ou égal à une valeur spécifiée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

INT(x)

Argument	Description
x	Valeur numérique.

### Exemple

INT(5,6) renvoie 5.

INT(-5.6) renvoie -6.

## ISUND

ISUND renvoie 1 si une valeur spécifiée n'est pas définie, autrement, renvoie 0.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

ISUND(x)

Argument	Description
x	Nombre ou expression.

### Exemple

ISUND(5,2) renvoie 0.

ISUND(1/0) renvoie 1.

## LN

LN renvoie le logarithme naturel (base e) d'un nombre.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

LN(x)



Argument	Description
x	Nombre positif. La fonction renvoie une erreur si x est négatif ou égal à zéro.

### Exemple

LN(10) renvoie 2,302585093.

## LOG

LOG renvoie le logarithme en base 10 d'un nombre positif.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

LOG(x)

Argument	Description
x	Nombre positif. La fonction renvoie une erreur si x est négatif ou égal à zéro.

### Exemple

LOG(10) renvoie 1.

## MAX

MAX renvoie la plus grande de deux valeurs.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

MAX(num1, num2)

Argument	Description
num1	Première de deux valeurs.
num2	Deuxième de deux valeurs.

### Exemple

MAX(10, 3) renvoie 10.

## MIN

MIN renvoie la plus petite de deux valeurs.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

MIN(num1, num2)

Argument	Description
num1	Première de deux valeurs.
num2	Deuxième de deux valeurs.

### Exemple

MIN(10, 3) renvoie 3.

## MOD

MOD renvoie le reste après la division d'un nombre par un autre nombre.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
MOD(number, divisor)
```

Argument	Description
number	Nombre dont vous recherchez le reste après la division.
divisor	Valeur par laquelle l'argument nombre est divisé.

### Exemple

MOD(10, 3) renvoie 1.

## RAND

RAND génère un nombre aléatoire distribué uniformément entre 0 et 1. Le générateur de nombres aléatoires est initialisé au chargement de TM1.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
RAND.
```

### Arguments

Aucun.

### Exemple

RAND génère un nombre aléatoire distribué uniformément entre 0 et 1

## ARRONDI

ROUND arrondit un nombre donné à l'entier le plus proche. L'arrondi peut être effectué de diverses manières.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

La forme d'arrondi élémentaire consiste à remplacer un nombre arbitraire par un entier. Un nombre y peut être arrondi en un entier q de plusieurs façons.

Les plus courantes sont les suivantes :

- **Arrondir vers l'entier le plus proche**

q est l'entier le plus proche de (voir "**Arrondir en partant de**" pour les règles de départage).

- **Arrondir vers zéro** (ou tronquer)

q est la partie entière de y, sans ses chiffres de fraction.

- **Arrondir vers le bas**

q est l'entier le plus grand ne dépassant pas y.

- **Arrondir vers le haut**

q est le plus petit entier non inférieur à y.

- **Arrondir en partant de zéro**

Si y est un entier, q est y ; autrement, q est l'entier le plus proche de 0 et tel que y est compris entre 0 et q.

TurboIntegrator utilise essentiellement la méthode d'**arrondi vers le bas** de  $\text{floor}(x + .5)$ . Microsoft Excel utilise la méthode d'**arrondi à l'entier le plus proche**. Il peut en résulter des entiers différents selon que vous utilisez un processus TurboIntegrator ou bien Excel.

### Syntaxe

ROUND(number)

Argument	Description
number	Nombre à arrondir.

### Exemple

ROUND(1,46) renvoie 1.

## ROUNDUP

ROUNDUP arrondit un nombre donné à une précision décimale spécifiée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

ROUNDUP(number, decimal)

Argument	Description
number	Nombre à arrondir.
decimal	Précision décimale à laquelle vous appliquez l'arrondissement. Si cet argument est positif, la fonction arrondit le nombre de décimales spécifié à droite de la virgule décimale. Si cet argument est négatif, la fonction arrondit le nombre de décimales spécifié à gauche de la virgule décimale.  L'argument de décimale doit être situé entre -15 et 15 ou être l'un de ces deux nombres.

### Exemple

ROUND(1,46, 1) renvoie 1,5.

ROUND(1,466, 2) renvoie 1,47.

ROUND(234.56, -1) renvoie 230,00.

ROUND(234.56, 0) renvoie 235,00.

## SIGN

SIGN détermine si un nombre est positif, négatif, ou nul. La fonction renvoie 1 si le nombre est positif, -1 s'il est négatif, ou 0 si le nombre est zéro.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

SIGN(number)

Argument	Description
number	Nombre.

### Exemple

SIGN(-2,5) renvoie -1.

## SIN

SIN renvoie le sinus d'un angle donné.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

SIN(x)

Argument	Description
x	Valeur, exprimée en radians, dont vous recherchez le sinus.

### Exemple

SIN(1,5708) renvoie 1.

## SQRT

SQRT renvoie la racine carrée d'une valeur donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

SQRT(x)

Argument	Description
x	Une valeur positive quelconque. SQRT renvoie une erreur si x est négatif.

### Exemple

SQRT(16) renvoie 4.

## TAN

TAN renvoie la tangente d'un angle donné.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

TAN(x)

Argument	Description
x	Valeur, exprimée en radians, dont vous recherchez la tangente.

### Exemple

TAN(0) renvoie 0.

TAN(,7854) renvoie 1.

## Fonctions de règles de texte

Règles destinées à gérer du texte dans les règles.

## CAPIT

CAPIT met en majuscule la première lettre de chaque mot d'une chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

CAPIT(string)

Argument	Description
string	Chaîne de texte.

### Exemple

CAPIT('ventes premier trimestre') renvoie 'Ventes Premier Trimestre'.

## CHAR

CHAR renvoie le caractère identifié par un code numérique ASCII donné.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

CHAR(number)

Argument	Description
number	Numéro de code ASCII. Ce numéro doit être entre 1 et 255, (1 et 255 étant compris).

## Exemple

CHAR(100) renvoie 'd'

## CODE

CODE renvoie le code numérique ASCII d'un caractère spécifié d'une chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

CODE(string, location)

Argument	Description
string	Chaîne de texte.
location	Numéro spécifiant le caractère de la chaîne dont vous recherchez la valeur de code ASCII.

## Exemple

CODE('321', 2) renvoie 50.

CODE('End', 3) renvoie 100.

## DELET

DELET renvoie le résultat de la suppression d'un nombre de caractères spécifié à partir d'un endroit spécifié d'une chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

DELET(string, start, number)

Argument	Description
string	Chaîne de texte.
start	Caractère à partir duquel commencer la suppression.
number	Nombre de caractères à supprimer.

### Exemple

DELET('paiement', 3, 3) renvoie 'paient'.

## FILL

FILL répète une chaîne donnée aussi longtemps que nécessaire pour renvoyer une chaîne d'une longueur spécifique.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
FILL(string, length)
```

Argument	Description
string	Chaîne de texte. Cette chaîne est répétée jusqu'à ce qu'elle atteigne la longueur spécifiée.
length	Longueur de la chaîne que la fonction doit renvoyer.

### Exemple

FILL('-', 5) renvoie '-----'.

FILL('ab', 5) renvoie 'ababa'.

## INSRT

INSRT insère une chaîne dans une autre à un point d'insertion spécifié.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
INSRT(string1, string2, location)
```

Argument	Description
string1	Chaîne de texte.
string2	Chaîne de texte.
location	Caractère dans string2 auquel vous voulez insérer string1. La fonction insère string1 dans string2 juste avant le caractère spécifié.

### Exemple

INSRT('ABC', 'DEF', 2) renvoie 'DABCEF'.

## LONG

LONG renvoie la longueur d'une chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

LONG(string)

Argument	Description
string	Chaîne de texte.

## Exemple

LONG('Ventes') renvoie 6.

## LOWER

LOWER convertit toutes les majuscules d'une chaîne en minuscules.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

LOWER(string)

Argument	Description
string	Chaîne de texte.

## Exemple

LOWER('Ventes Premier Trimestre') renvoie 'ventes premier trimestre'.

## NUMBR

NUMBR convertit une chaîne en nombre. La chaîne passée à la fonction NUMBR doit utiliser. (point) comme séparateur décimal et , (virgule) comme séparateur de milliers. Tout autre signe décimal/ séparateur de milliers produit des résultats incorrects.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

NUMBR(string)

Argument	Description
string	Chaîne à convertir en nombre. Tous les caractères autres que '0' à '9', '+', '-', '.', et 'E' sont ignorés.

## Exemple

NUMBR('-5.6') renvoie -5.6.

NUMBR('-5A. B6C') renvoie -5,6.

## SCAN

SCAN renvoie un nombre indiquant le point de départ de la première occurrence d'une sous-chaîne spécifiée dans une chaîne. Si la sous-chaîne n'apparaît pas dans une chaîne donnée, la fonction renvoie 0.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.



## Syntaxe

```
SCAN(substring, string)
```

Argument	Description
substring	Sous-chaîne à rechercher.
string	Chaîne dans laquelle vous recherchez la sous-chaîne.

Les arguments de cette fonction sont sensibles à la casse. Pour que la fonction renvoie une valeur différente de zéro, l'usage des majuscules dans l'argument **substring** doit correspondre exactement à celui utilisé dans l'argument **string**.

## Exemple

SCAN('scribe', 'described') renvoie 3.

Toutefois, SCAN('Scribe', 'described') renvoie 0 car la casse de l'argument **substring** (Scribe) ne correspond pas à la casse de l'argument **string** (described).

## STR

STR convertit un nombre en chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

## Syntaxe

```
STR(number, length, decimal)
```

Argument	Description
number	Nombre à convertir en chaîne. Ce nombre peut contenir un signe positif ou négatif. Il peut contenir des décimales.

Argument	Description
length	<p>Longueur de la chaîne dans le résultat de la fonction, comprenant le signe, les séparateurs, les décimales ou les positions décimales.</p> <p>Si le nombre de chiffres dans le nombre est inférieur à la valeur du paramètre length, la fonction peut insérer des espaces vides au début pour atteindre cette longueur après l'insertion du signe, des séparateurs, des décimales ou des positions décimales.</p> <p>Si le nombre de chiffres dans le nombre dépasse la valeur du paramètre length, il est tronqué par la fonction pour atteindre cette longueur.</p> <p>Si le paramètre length n'est pas spécifié, ce doit être un nombre positif supérieur à "0".</p> <p>S'il n'est pas spécifié, le paramètre number est conservé et la longueur du résultat de la fonction inclut le signe, les séparateurs, les décimales et les positions décimales.</p>
decimal	<p>Nombre de décimales à inclure dans le résultat de la fonction.</p> <p>Si ce paramètre a la valeur "0" ou n'est pas spécifié, un séparateur décimal n'est pas inclus.</p> <p>Si le nombre indiqué comporte plus de décimales que demandé, le résultat de la fonction est arrondi.</p>

### Exemples

Appel de fonction	Numérique	Longueur	Décimal	Résultat
STR(3.14159, 6, 2)	3.14159	6	2	" 3.14"
STR(-3.14, 6, 0)	-3.14159	6	0	" -3"
STR(3.14159,5,3)	3.14159	5	3	"3.142"
STR(1000000,4,0)	1000000	4	0	"1000"
				Notez que le nombre est tronqué.
STR(1000000,4,2)	1000000	4	2	"1000000.00"
				Notez que le nombre n'est pas tronqué car une décimale est spécifiée.
STR(10,2,4)	10	2	4	"10.0000"
STR(1000000,5,0)	1000000	5	0	"1,000"

Appel de fonction	Numérique	Longueur	Décimal	Résultat
STR(1000000,10,5 )	1000000	10	5	<p>Notez que le nombre est tronqué. Ceci est la valeur renvoyée si des séparateurs sous forme de virgule sont spécifiés.</p> <p>"1,000,000.00000"</p> <p>Ceci est la valeur renvoyée si des séparateurs sous forme de virgule sont spécifiés.</p>

## SUBST

SUBST renvoie une sous-chaîne d'une chaîne donnée.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
SUBST(string, beginning, length)
```

Argument	Description
string	Chaîne dont vous voulez extraire la sous-chaîne.
beginning	Caractère auquel commence la sous-chaîne.
length	Longueur de la sous-chaîne.

### Exemple

SUBST('Retraite', 3, 4) renvoie 'trait'.

## TRIM

TRIM renvoie le résultat de la suppression des espaces blancs précédant ou suivant une chaîne.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
TRIM(string)
```

Argument	Description
string	Chaîne de texte.

### Exemple

TRIM(' Premier trimestre ') renvoie 'Premier trimestre'.

## UPPER

UPPER convertit une chaîne de texte en majuscules.

Cette fonction est valide dans les règles et les processus TM1 TurboIntegrator.

### Syntaxe

```
UPPER(string)
```

Argument	Description
string	Chaîne de texte.

### Exemple

UPPER('Résultats premier trimestre') renvoie RESULTATS PREMIER TRIMESTRE.

## Fonctions de règles diverses

Fonctions de règles introuvables dans d'autres catégories.

### FEEDERS

Lorsque vous utilisez une déclaration SKIPCHECK pour restaurer des consolidations de faible volume dans une règle TM1, assurez-vous que toutes les cellules calculées par des règles sont identifiées par des instructions FEEDERS. A cet effet, insérez une déclaration FEEDERS tout de suite après toutes les instructions de règles :

```
FEEDERS;
```

Tout de suite après l'instruction FEEDERS, créez des instructions feeder identifiant les cellules calculées par des règles dans le cube.

Pour une description détaillée des règles TM1, y compris les consolidations à faible densité et la création de FEEDERS, reportez-vous au document *TM1 Rules*.

### FEEDSTRINGS

Les valeurs de chaîne générées par des règles ne sont pas affichées quand les zéros sont supprimés dans une vue, sauf quand la chaîne réside dans une cellule alimentée. Pour activer l'alimentation des cellules de chaîne, insérez l'instruction FEEDSTRINGS comme première ligne de votre règle.

```
FEEDSTRINGS;
```

Une fois cette instruction insérée, vous pouvez définir des feeders pour les cellules de chaîne dans une vue de cube et être assuré que la chaîne sera disponible pour les autres règles même si la vue est une vue dans laquelle les zéros ont été supprimés. Les instructions qui définissent des feeders pour les cellules de chaîne doivent être créées sous l'instruction FEEDERS de votre règle.

Comme dans le cas de feeders numériques, l'alimentation d'une cellule consolidée entraîne l'alimentation de tous les composants de la consolidation. Etant donné que vous pouvez stocker les chaînes dans des cellules consolidées, soyez extrêmement vigilant si ces cellules sont utilisées pour alimenter d'autres cellules. Une utilisation excessive de feeders peut entraîner une multiplication exagérée des calculs et une performance insatisfaisante de l'application.

Pour une description détaillée des règles TM1, y compris la création de feeders, reportez-vous au document *TM1 Rules*.

## SKIPCHECK

Vous pouvez restaurer les consolidations à faible densité et améliorer la performance en insérant une instruction SKIPCHECK au début de la règle TM1.

Lors des consolidations, TM1 utilise un algorithme de consolidation à faible densité pour sauter les cellules vides ou contenant un zéro. Cet algorithme accélère les calculs de consolidation dans les cubes à très faible densité. Un cube à faible densité est un cube dans lequel le pourcentage de cellules chargées par rapport au nombre total de cellules est faible.

Lors de la consolidation de données dans des cubes pour lesquels des règles ont été définies, TM1 désactive cet algorithme de consolidation à faible densité vu qu'une ou plusieurs cellules vides sont peut-être calculées à l'aide d'une règle. Le fait d'ignorer des cellules calculées par des règles entraîne des totaux consolidés incorrects. Quand un algorithme de consolidation de faible densité est désactivé, le programme vérifie s'il y a une valeur dans chaque cellule pendant la consolidation. Ceci peut ralentir les calculs dans les cubes très volumineux et à faible densité.

```
SKIPCHECK;
```

Si une règle comprend une instruction FEEDSTRINGS, l'instruction SKIPCHECK doit être la deuxième instruction de la règle. Si la règle ne comprend pas d'instruction FEEDSTRINGS, l'instruction SKIPCHECK doit être la première instruction de la règle.

Quand vous utilisez la fonction SKIPCHECK pour restaurer des consolidations de faible volume, assurez-vous aussi que la règle comprend une instruction et que toutes les cellules calculées par des règles sont identifiées par des instructions FEEDERS.

Pour une description détaillée des règles TM1, y compris les consolidations à faible densité et la création de FEEDERS, reportez-vous au document *TM1 Rules*.



---

## Chapitre 3. Fonctions de macros

IBM Planning Analytics inclut un ensemble de fonctions macro que vous pouvez incorporer dans des macros Microsoft Excel. Vous pouvez utiliser ces fonctions macro dans TM1 Perspectives pour accéder aux serveurs, aux données et structures de cube, ainsi qu'aux options TM1.

**Remarque :** Avant d'exécuter ces macros, vous devez charger le module d'extension TM1 (Tm1p.xla). Pour obtenir des informations sur le chargement des extensions, voir l'aide de Microsoft Excel.

---

### Accès aux fonctions macro de Microsoft Excel 2010 et versions ultérieures

#### Procédure

1. Cliquez avec le bouton droit de la souris sur l'onglet de feuille de la feuille de calcul active.
2. Dans le menu des raccourcis, cliquez sur **Insérer**.
3. Double-cliquez sur **Macro MS Excel 4.0**.
4. Cliquez sur la cellule dans laquelle vous voulez insérer la fonction macro.
5. Cliquez sur **Formules**, puis sur **Insérer une fonction**.
6. Dans la liste de catégories, sélectionnez **TM1**.
7. Sélectionnez la fonction à insérer, puis cliquez sur **OK**.
8. Entrez les valeurs des arguments.
9. Cliquez sur **OK** pour insérer la fonction dans la cellule en cours de la feuille de macro.

---

### Accès aux fonctions de macros dans les modules VBA

Pour accéder aux fonctions macro à partir des modules VBA, utilisez la méthode Exécuter.

```
Run ("macro_function", arg1, ...)
```

#### Exemple

```
Sub Elemlist( )  
    Worksheets("Sheet1").Select  
    Cells(3,5).Select  
    ActiveCell.Value = Run ("E_PICK", "local:Region")  
End Sub
```

Cette procédure appelle la fonction de macro E\_PICK qui accède à une liste éléments de la dimension Region. L'élément sélectionné remplit une cellule de la feuille 1.

---

### D\_PICK

Cette fonction appelle une boîte de dialogue qui répertorie toutes les dimensions disponibles dans le répertoire de données local et sur les serveurs distants connectés. La dimension que vous sélectionnez dans la boîte de dialogue devient la valeur de la fonction D\_PICK.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

#### Syntaxe

```
D_PICK
```

## Arguments

Aucun.

## D\_FSAVE

Cette fonction permet de créer ou de mettre à jour de très grandes dimensions dont le nombre de lignes des feuilles de dimension dépasserait le maximum permis par sur une feuille de calcul Excel.

Pour utiliser la fonction D\_FSAVE, créez un fichier ASCII à séparateurs nommé dim.dit, dim étant le nom d'une dimension à créer ou mettre à jour. Ce fichier doit résider dans le répertoire de données du serveur local.

La structure du fichier ASCII doit correspondre à celle d'une feuille de dimension, comme suit :

- Insérez trois zones par ligne.
- Dans la première zone, spécifiez le type d'élément : C pour consolidation, N pour numérique, S pour chaîne (String), et vide pour un composant de consolidation.
- Dans la deuxième zone, spécifiez le nom de l'élément.
- Dans la troisième zone, spécifiez la pondération, le cas échéant. La pondération par défaut est 1,0.

Séparez les zones à l'aide du délimiteur défini dans votre système d'exploitation. Dans Windows, ce délimiteur est défini par l'entrée Délimiteur de liste dans la boîte de dialogue Propriétés des paramètres régionaux.

En cas d'erreurs dans la structure du fichier ASCII, par exemple éléments mal placés ou indéfinis, le programme affiche un message d'erreur.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
D_FSAVE(file)
```

Argument	Description
file	Nom d'un fichier ASCII à séparateurs dont l'extension est .dit. N'ajoutez pas d'extension de fichier. Ce fichier doit résider dans votre répertoire de données TM1 local.

## Exemple

```
=D_FSAVE("Region")
```

Cet exemple lit un fichier ASCII appelé Region.dit et crée ou met à jour la dimension Region.

**Remarque :** D\_FSAVE peut servir à créer ou mettre à jour des dimensions sur des serveurs distants. Toutefois, cette fonction cherche toujours le fichier.dit dans le répertoire de données local (tel que défini dans Tm1p.ini). Assurez-vous que le fichier .dit de la dimension à créer/mettre à jour réside dans le répertoire de données local, puis spécifiez le serveur sur lequel vous voulez créer/mettre à jour la dimension en utilisant le nom de serveur comme préfixe de fichier .dit.

```
=D_FSAVE("ServTM1:Region")
```

Cet exemple recherche un fichier nommé région.dit dans le répertoire de données du serveur local, puis écrit la dimension Region dans le répertoire de données du serveur ServTM1.



## D\_SAVE

Cette fonction enregistre la feuille active comme fichier de feuille de dimensions (dim.xdi). Le nom du classeur est utilisé comme nom de fichier. Ensuite, TM1 crée ou met à jour la dimension spécifiée par le nom du classeur.

Si la feuille de calcul active n'est pas conforme au format de feuille de dimension, ou si des informations manquent, un message d'erreur s'affiche. Vous devez, par exemple, définir tous les éléments utilisés dans une consolidation de niveau 1 comme éléments numériques (N).

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
D_SAVE
```

### Arguments

Aucun.

## DBProportionalSpread

Cette fonction répartit une valeur spécifiée aux feuilles d'une consolidation proportionnellement aux valeurs de cellule existantes.

Cette fonction est analogue à la méthode de répartition de données dite Répartition proportionnelle qui est décrite en détail dans la documentation *TM1 Perspectives and TM1 Architect*.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
DBProportionalSpread( value, server:cube, e1, e2, e3...,  
e16 )
```

Argument	Description
value	Valeur à répartir.
server:cube	Nom du cube, précédé du nom du serveur approprié, dans lequel vous voulez répartir la valeur.  Par exemple, pour répartir les valeurs dans le cube Ventes sur le serveur Comptabilité, vous spécifiez Comptabilité:Ventes.
e1...e16	Noms des éléments identifiant la consolidation dont les feuilles accepteront la valeur répartie.

### Exemple

```
DBProportionalSpread(2000, "Comptabilité:Ventes", "Réal", "Argentine", "Berline 1,8L Série S",  
"Ventes", "Trim 1")
```

Cet exemple répartit la valeur 2000 aux enfants de la consolidation identifiée par les éléments Réel, Argentine, Berline 1,8L Série S, Ventes et Trim1. Les valeurs sont réparties dans le cube Ventes sur le serveur Comptabilité.

## E\_PICK

Cette fonction appelle l'Editeur de sous-ensemble et génère la liste de tous les éléments de la dimension spécifiée. Le nom d'élément que vous sélectionnez dans l'Editeur de sous-ensemble devient la valeur renvoyée par la fonction E\_PICK.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
E_PICK(Dimension, Alias, Subset, Element)
```

Argument	Description
Dimension	Nom de dimension valide. La dimension peut résider dans le répertoire de données local ou sur un serveur distant auquel vous êtes connecté.  Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:dim. Pour un serveur distant, spécifiez servername:dim.
Alias	Nom d'un alias existant pour le sous-ensemble. Quand ce paramètre est défini, l'alias est appliqué lorsque le sous-ensemble est ouvert dans l'Editeur de sous-ensemble et la fonction renvoie l'alias de l'élément sélectionné.  Si vous choisissez de ne pas définir un paramètre Alias, vous devez passer une chaîne vide à la fonction.
Subset	Nom du sous-ensemble à ouvrir dans l'Editeur de sous-ensemble quand la fonction E_PICK est appelée. Le paramètre Alias doit être fourni pour utiliser ce paramètre. Le paramètre Alias peut être défini comme chaîne vide ("").  Si vous choisissez de ne pas définir un paramètre Sous-ensemble, vous devez passer une chaîne vide à la fonction.
ElementNameOrIndex	Nom ou numéro d'index de l'élément à présélectionner lors de l'ouverture de l'Editeur de sous-ensemble.  Si vous choisissez de ne pas définir un paramètre ElementNameOrIndex, vous devez passer une chaîne vide à la fonction.

### Exemple 1

```
=E_PICK("TM1SERV:Region"," "," "," ")
```

Cet exemple ouvre la dimension Region dans l'Editeur de sous-ensemble.

```
=E_PICK ("TM1SERV:Region","Deutsch","Europe","Argentina")
```

Cet exemple ouvre le sous-ensemble Europe dans l'Editeur de sous-ensemble. L'alias Deutsche est appliqué et l'élément Argentina est présélectionné quand l'Editeur de sous-ensemble s'ouvre.

```
=E_PICK ("TM1SERV:Region"," "," ",14)
```

Cet exemple ouvre la dimension Region dans l'Editeur de sous-ensemble et le 14ème élément dans la définition de dimension est présélectionné.

## I\_EXPORT

Cette fonction exporte des données du cube spécifié dans un fichier ASCII délimité, qui est créé dans le répertoire Mes documents de l'utilisateur en cours. Dans la plupart des cas, le répertoire Mes documents est C:\Users\\Documents.

**Remarque :** I\_EXPORT applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction pour exporter un gros cube, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
I_EXPORT(cube, file, zero, calcs)
```

Argument	Description
cube	Nom de cube valide. Le cube peut résider dans le répertoire de données local ou sur un serveur distant auquel vous êtes connecté. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
file	Nom du fichier ASCII délimité à créer. Ce fichier aura l'extension .cma, mais vous ne devez pas la spécifier.
zero	Spécifie si les valeurs nulles seront incluses. Spécifiez TRUE pour les inclure ou FALSE pour les exclure.
calcs	Spécifie si les valeurs calculées seront incluses. Spécifiez TRUE pour les inclure ou FALSE pour les exclure.

### Exemple

```
=I_EXPORT("local:92réel4d","Charger",FALSE,TRUE)
```

Cet exemple exporte les données depuis le cube 92réel4d vers le fichier Charger.cma. Les valeurs nulles sont exclues et les valeurs calculées sont incluses.

## I\_NAMES

Vous pouvez utiliser I\_NAMES pour créer une liste de noms d'élément. Cette fonction lit un fichier ASCII à séparateur et écrit tous les noms uniques d'une colonne spécifiée dans la colonne correspondante de la feuille de calcul active.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
I_NAMES(file, column)
```

Argument	Description
file	Nom d'un fichier ASCII à séparateurs dont l'extension est .cma. N'ajoutez pas d'extension de fichier.
column	Nombre spécifiant à la fois la zone du fichier ASCII depuis lequel les noms doivent être lus et la colonne de la feuille de calcul active dans laquelle ces noms doivent être écrits.

### Exemple

```
=I_NAMES("Ventes98",3)
```

Cet exemple lit le fichier Ventes98.cma. Tous les noms uniques de la troisième colonne sont écrits dans la colonne C de la feuille de calcul active.

## I\_PROCESS

Cette fonction écrit l'un après l'autre les enregistrements d'un fichier ASCII, sur la première ligne de la feuille de calcul active. Chaque zone remplit une cellule différente. La feuille de calcul est recalculée après l'écriture de chaque enregistrement.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
I_PROCESS(file)
```

Argument	Description
file	Nom d'un fichier ASCII à séparateurs dont l'extension est .cma. N'ajoutez pas d'extension de fichier.

### Exemple

```
=I_PROCESS("Ventes98")
```

Cet exemple écrit chaque enregistrement du fichier Ventes98.cma dans la première ligne de la feuille de calcul active.

## M\_CLEAR

Cette fonction efface et recharge toutes les dimensions en mémoire. Elle n'efface pas les cubes et ne redémarre pas le serveur.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
M_CLEAR
```

### Arguments

Aucun.

## N\_CONNECT

Cette fonction établit une connexion à un serveur distant. Si la connexion réussit, N\_CONNECT ne renvoie aucune valeur. Si la connexion est impossible, des messages d'erreur de serveur sont renvoyés.

**Remarque :** la fonction N\_CONNECT n'est pas prise en charge lorsqu'un serveur utilise la connexion intégrée ou la sécurité IBM Cognos8 pour l'authentification. Cette fonction peut uniquement se connecter à un serveur configuré pour utiliser l'authentification TM1 standard.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
N_CONNECT(server, client, password)
```

Argument	Description
server	Nom d'un serveur distant. Ce serveur doit être enregistré sur le serveur Admin auquel votre client fait référence.
client	Nom d'utilisateur qui se connecte au serveur spécifié.
password	Mot de passe du client spécifié.

### Exemple

```
=N_CONNECT("Ventes", "USR2", "Espadon")
```

Cet exemple établit une connexion au serveur distant Sales en utilisant le nom de client USR2 et le mot de passe Espadon.

## N\_DISCONNECT

Cette fonction vous déconnecte de tous les serveurs distants auxquels vous êtes connectés. La fonction ne vous déconnecte pas de votre serveur local. N\_DISCONNECT renvoie TRUE si elle parvient à vous déconnecter de tous les serveurs auxquels vous êtes connecté. Elle renvoie FALSE si vous ne pouvez pas vous déconnecter d'un serveur distant.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

N\_DISCONNECT

## Arguments

Aucun.

## OPTGET

---

Cette fonction renvoie la valeur en cours d'une option dans le fichier Tm1p.ini.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

OPTGET(option)

Argument	Description
option	Nom d'option TM1 valide.

Valeurs d'option valides	Description
AdminHost	Renvoie le nom ou l'adresse de l'hôte Admin auquel votre client fait référence.
AnsiFiles	Renvoie T si le jeu de caractères ANSI est actuellement utilisé pour importer des données de fichiers ASCII à séparateurs. Renvoie F si le jeu de caractères ASCII est actuellement utilisé.
DataBaseDirectory	Renvoie le chemin d'accès complet au répertoire de données du serveur local.
GenDBRW	Renvoie F si la feuille de tranche contient des formules DBR. Renvoie T si la feuille de tranche contient des formules DBRW.
NoChangeMessage	Renvoie T si cette option est définie pour renvoyer le message NO CHANGE quand une formule DBSn pointe à une cellule de niveau C. Renvoie F si cette option est définie sur F.

## Exemple

```
=OPTGET("DataBaseDirectory")
```

Cet exemple renvoie le chemin d'accès complet au répertoire de données du serveur local.

## OPTSET

---

OPTSET définit une valeur pour une option TM1 spécifiée.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
OPTSET(option, value)
```

Argument	Description
option	Nom d'option TM1 valide.
value	Valeur valide pour l'option spécifiée.

Valeurs d'option valides	Description
AdminHost	Spécifie le nom de l'hôte Admin où tourne un serveur Admin.
AnsiFiles	Spécifie une valeur définissant le jeu de caractères utilisé lors des importations de données. Spécifiez T pour utiliser le jeu de caractères ANSI. Spécifiez F pour utiliser le jeu de caractères ASCII.
DataBaseDirectory	Spécifiez une valeur définissant le chemin d'accès complet au répertoire de données du serveur local.
GenDBRW	Spécifiez une valeur déterminant la formule à utiliser par TM1 pour lier les valeurs des feuilles de tranche à des cubes. Spécifiez T pour générer des formules DBRW à la création des feuilles de tranche. Spécifiez F pour générer des formules DBR.
NoChangeMessage	Spécifiez une valeur déterminant si TM1 affiche le message NO CHANGE lorsqu'une formule DBSn pointe vers une cellule de niveau C. Spécifiez T pour afficher le message. Spécifiez F pour n'afficher que la valeur.

## Exemple

```
=OPSET("DataBaseDirectory", "c:\Tm1data")
```

Cet exemple définit le répertoire de données local à c:\Tm1data.

## PublishSubset

Cette fonction publie un sous-ensemble privé nommé sur un serveur. Si vous essayez de publier un sous-ensemble privé pour lequel un sous-ensemble public de nom identique existe, l'application vous invite à remplacer le sous-ensemble public existant.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
PublishSubset(dimension, subset)
```

Argument	Description
dimension	Nom de la dimension (précédé du préfixe du serveur) contenant le sous-ensemble privé à publier. Par exemple, pour publier un sous-ensemble de la dimension Region sur le serveur Finances, vous saisissez 'Finance:Region' comme argument de la dimension.
subset	Nom du sous-ensemble privé à publier.

## PublishView

Cette fonction publie une vue privée nommée sur un serveur. Elle ne peut publier de vue privée utilisant des sous-ensembles privés.

Tous les sous-ensembles privés d'une vue privée doivent d'abord être publiés avec la fonction de macros PublishSubset. Si vous tentez de publier une vue privée pour laquelle une vue publique de nom identique existe, l'application vous invitera à remplacer la vue publique existante.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
PublishView(cube, view)
```

Argument	Description
cube	Nom du cube (avec préfixe du serveur) contenant la vue privée à publier. Par exemple, pour publier une vue du cube Projections sur le serveur Finances, vous passez "Finance:Projections" comme argument de cube.
view	Nom de la vue privée à publier.

## QUDEFINE

QUDEFINE définit et sauvegarde les paramètres des ensembles de requêtes TM1. QUDEFINE équivaut à créer un ensemble de requêtes dans la boîte de dialogue Extraction de vue.

Les requêtes créées avec cette fonction peuvent être exécutées dans la boîte de dialogue Extraction de vue.

Vous pouvez également utiliser l'ensemble des requêtes comme argument des fonctions de macros QUEXPORT, QULoop et QUSUBSET.

**Remarque :** QUDEFINE applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction créer une requête qui couvre une section importante d'un cube, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.



## Syntaxe

```
QUDEFINE(cube, query, range, LowLim, HiLim, SkpZeroes,  
SkpCons)
```

Argument	Description
cube	Nom du cube sur lequel portera la requête.  Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
query	Nom de la requête à enregistrer pour usage futur.
range	<p>Plage de cellules de feuille de calcul comprenant une colonne pour chaque dimension du cube. Lorsque vous exécutez la requête, TM1 n'examine que les cellules de cube identifiées par les éléments spécifiés ou auxquels la plage fait référence.</p> <p>La plage doit contenir une colonne pour chaque dimension du cube. L'ordre des colonnes doit être le même que celui des dimensions du cube.</p> <p>Dans chaque colonne, vous spécifiez ou faites référence aux éléments à inclure. Pour inclure un sous-ensemble d'éléments, énumérez les noms d'élément ou spécifiez un nom de sous-ensemble. Faites précéder le nom du sous ensemble d'une barre oblique inverse (\). Par exemple, \quarter spécifie le sous-ensemble 'quarter' (trimestre). Pour inclure tous les éléments d'une dimension (le sous-ensemble TOUT), laissez cette colonne vide.</p> <p>Vous pouvez utiliser les fonctions DBR pour remplir les cellules de la plage. Si cette fonction renvoie des valeurs vides pour toute colonne de la rangée, QUDEFINE utilise le sous-ensemble TOUT pour la dimension associée à cette colonne.</p>
LowLim	Valeur de cellule la plus basse à considérer pour l'exportation.
HighLim	Valeur de cellule la plus haute à considérer pour l'exportation.
SkpZeroes	Spécifie si les cellules contenant des zéros sont ignorées. Spécifiez TRUE pour les exclure ou FALSE pour les inclure.
SkpCons	Spécifie si les cellules contenant des valeurs consolidées sont ignorées. Spécifiez TRUE pour les exclure ou FALSE pour les inclure.

## Exemple

```
=QUDEFINE("local:Ventes98", "Meilleures_ventes", Feuille1!B3:F5, 3000, 5000, TRUE, TRUE)
```

Cet exemple crée une requête contenant les éléments énumérés dans la feuille Feuille1, dans la plage de cellules B3:F5. Quand vous exécutez cette requête, TM1 n'inspecte que les cellules de cube identifiées par ces éléments et exporte les valeurs comprises entre 3 000 et 5 000 qui ne sont pas calculées.

**Remarque :** Si liminf ou limsup est une chaîne comprenant des caractères numériques, Excel exige que cette chaîne soit entre une série de quatre guillemets doubles et esperluettes simples, comme dans l'exemple suivant :

```
""""&"0123"&""""
```

## QUDEFINEEX

QUDEFINEEX définit et sauvegarde les paramètres des ensembles de requêtes TM1. Elle équivaut à créer un ensemble de requêtes dans la boîte de dialogue Extraction de vue. Cette fonction est identique à la macro QUDEFINE à l'exception du fait que QUDEFINEEX comprend un argument permettant d'exclure de la requête les valeurs dérivées de règles.

Les requêtes créées avec cette fonction peuvent être exécutées dans la boîte de dialogue Extraction de vue.

Vous pouvez également utiliser l'ensemble des requêtes comme argument des fonctions de macros QUEXPORT, QULoop et QUSUBSET.

**Remarque :** QUDEFINEEX applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction créer une requête qui couvre une section importante d'un cube, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
QUDEFINEEX(cube, query, range, lowlim, hilim, skpZeroes,  
skpCons, skpRuleVals)
```

Argument	Description
cube	Nom du cube sur lequel portera la requête.
	Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
query	Nom de la requête à enregistrer pour usage futur.

Argument	Description
range	<p>Plage de cellules de feuille de calcul comprenant une colonne pour chaque dimension du cube. Lorsque vous exécutez la requête, TM1 n'examine que les cellules de cube identifiées par les éléments spécifiés ou auxquels la plage fait référence.</p> <p>La plage doit contenir une colonne pour chaque dimension du cube. L'ordre des colonnes doit être le même que celui des dimensions du cube.</p> <p>Dans chaque colonne, vous spécifiez ou faites référence aux éléments à inclure. Pour inclure un sous-ensemble d'éléments, énumérez les noms d'élément ou spécifiez un nom de sous-ensemble. Faites précéder le nom du sous ensemble d'une barre oblique inverse (\). Par exemple, \quarter spécifie le sous-ensemble 'quarter' (trimestre). Pour inclure tous les éléments d'une dimension (le sous-ensemble TOUT), laissez cette colonne vide.</p> <p>Vous pouvez utiliser les fonctions DBR pour remplir les cellules de la plage. Si cette fonction renvoie des valeurs vides pour toute colonne de la plage, QUDEFINEEX utilise le sous-ensemble TOUT pour la dimension associée à cette colonne.</p>
lowlim	Valeur de cellule la plus basse à considérer pour l'exportation.
highlim	Valeur de cellule la plus haute à considérer pour l'exportation.
skpZeroes	Spécifie si les cellules contenant des zéros sont ignorées. Spécifiez TRUE pour les exclure ou FALSE pour les inclure.
skpCons	Spécifie si les cellules contenant des valeurs consolidées sont ignorées. Spécifiez TRUE pour les exclure ou FALSE pour les inclure.
skpRuleVals	Spécifie si les cellules contenant des valeurs dérivées de règles sont ignorées. Spécifiez TRUE pour les exclure ou FALSE pour les inclure.

### Exemple

```
=QUDEFINEEX("local:Ventes", "Meilleures_ventes", Feuille1!B3:F5, 3000, 5000, TRUE, FALSE)
```

Cet exemple crée une requête contenant les éléments énumérés dans la feuille Feuille1, dans la plage de cellules B3:F5. Quand vous exécutez cette requête, TM1 n'inspecte que les cellules de cube identifiées par ces éléments et exporte les valeurs allant de 3 000 à 5 000 qui ne sont pas calculées, y compris celles qui sont dérivées de règles.

**Remarque :** Si liminf ou limsup est une chaîne comprenant des caractères numériques, Excel exige que cette chaîne soit entre une série de quatre guillemets doubles et esperluettes simples, comme dans l'exemple suivant :

```
""""&"0123"&""""
```

## QUEXPOR

Cette fonction exporte des valeurs de cellule depuis le cube spécifié vers un fichier ASCII délimité.

Pour créer la requête, servez-vous de la fonction QUDEFINE.

Chaque sortie d'enregistrement a le format suivant :

- Le nom du cube contenant les valeurs exportées.
- Le nom des éléments identifiant l'emplacement de cellule d'une valeur exportée unique.
- La valeur exportée

Pour un cube à cinq dimensions, TM1 crée des enregistrements contenant sept zones :

```
"nom de cube", "élé1", "élé2", "élé3", "élé4", "élé5", valeur
```

**Remarque :** QUEXPOR applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction pour exporter des valeurs d'une requête importante, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
QUEXPOR(cube, query, file)
```

Argument	Description
cube	Nom du cube sur lequel portera la requête. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
query	Nom d'une requête existante.
file	Nom du fichier ASCII (.cma) à séparateurs qui contiendra les données de cube exportées. N'ajoutez pas d'extension de fichier. Ce fichier est créé dans le répertoire de données local.

### Exemple

```
=QUEXPOR("ventes:Ventes98", "Camionnettes", "Camionnettes")
```

Cet exemple exporte les données du cube Ventes98 à l'aide de la requête Berlines. Les enregistrements sont écrits dans le fichier Sedans.cma.

## QULOOP

Cette fonction exporte des données satisfaisant aux critères de l'ensemble de requêtes à partir du cube spécifié. TM1 lit chaque enregistrement en sortie, l'un après l'autre, dans la première ligne de la feuille active. Chaque zone remplit une cellule différente. La feuille de calcul est recalculée après l'écriture de chaque enregistrement.

Chaque sortie d'enregistrement a le format suivant :

- Le nom du cube contenant les valeurs exportées.
- Le nom des éléments identifiant l'emplacement de cellule d'une valeur exportée unique.
- La valeur exportée

Pour un cube à cinq dimensions, TM1 crée des enregistrements contenant sept zones :

```
"nom de cube", "élé1", "élé2", "élé3", "élé4", "élé5", valeur
```

Servez-vous de QULOOP avec une formule DBSn pour remplir les cellules d'un cube.

**Remarque :** QULOOP applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction pour exporter des valeurs d'une requête importante, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
QULOOP(cube, query)
```

Argument	Description
cube	Nom du cube sur lequel portera la requête. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
query	Nom d'une requête existante.

### Exemple

```
=QULOOP("ventes:ventes98", "Berlines")
```

Cet exemple exporte les données du cube Ventes98 à l'aide de la requête Berlines.

## QUSUBSET

Cette fonction équivaut à exécuter une requête depuis la boîte de dialogue Extraction de vue appelée de l'Editeur de sous-ensemble.

**Remarque :** QUSUBSET applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction exécuter une requête qui renvoie un grand nombre d'éléments, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
QUSUBSET(cube, query, dimension, subset)
```

Argument	Description
cube	Nom du cube sur lequel portera la requête. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
query	Nom d'une requête existante.
dimension	Nom d'une dimension pour laquelle la requête existe.
subset	Nom du sous-ensemble de dimension à créer ; ce sous-ensemble contiendra tous les éléments satisfaisant aux critères de requête du sous-ensemble. Par exemple, un sous-ensemble peut renvoyer une liste des régions dans lesquelles les ventes de véhicules automobiles dépassent un montant spécifié.

## Exemple

```
=QUSUBSET("sales:Ventres98sales", "Meilleures", "Région", "Meilleures ventes")
```

Cet exemple crée le sous-ensemble Meilleures ventes pour la dimension Region à partir des critères de la requête Meilleures.

## R\_SAVE

Cette fonction enregistre la feuille active comme feuille de règles et la compile en un fichier .rux. Le classeur doit avoir le même nom que le cube pour lequel les règles sont compilées.

Toute instruction empêchant la compilation de ces règles est écrite dans le fichier tm1erlog.cma file dans le répertoire de données local.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
RSAVE
```

## Arguments

Aucun.

## SUBDEFINE

Cette fonction crée un sous-ensemble de dimension constitué de noms d'élément trouvés dans la feuille de calcul active.

Lorsque SUBDEFINE crée le sous-ensemble, celui-ci est créé en tant que sous-ensemble privé.

Si le sous-ensemble nommé existe déjà en tant que sous-ensemble privé lors de l'exécution de la fonction, le nouveau sous-ensemble écrasera celui existant sous ce nom.

Si le sous-ensemble nommé existe déjà en tant que sous-ensemble public, SUBDEFINE crée néanmoins le sous-ensemble en tant que sous-ensemble privé. Si vous désirez écraser le sous-ensemble public nommé existant, vous devrez pour ce faire publier le sous-ensemble privé créé par la fonction SUBDEFINE.

**Remarque :** SUBDEFINE applique un verrou au serveur, empêchant ainsi d'autres utilisateurs d'y accéder lors de l'exécution de la fonction. Si vous utilisez cette fonction pour créer un sous-ensemble avec un grand nombre d'éléments, il se peut que le serveur soit inaccessible pendant un certain temps.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
SUBDEFINE(dimension, subset, range)
```

Argument	Description
dimension	Nom de la dimension pour laquelle vous voulez créer un sous-ensemble. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:dim. Pour un serveur distant, spécifiez nomserveur:dim.
subset	Nom du sous-ensemble de dimension.
range	Plage de cellules de feuille de calcul contenant les noms des éléments de la dimension. Toute valeur de cellule de la plage ne représentant pas un élément valide est ignorée.

## Exemple

```
=SUBDEFINE("local:Modèle", "Smith", B7:M7)
```

Cet exemple crée un sous-ensemble appelé Smith pour la dimension Modèle. Le sous-ensemble contient des éléments se trouvant dans la plage de cellules B7:M7.

## SUBPICK

Cette fonction appelle une boîte de dialogue répertoriant tous les éléments du sous-ensemble spécifié. Les éléments que vous sélectionnez sont insérés dans la feuille de calcul active à partir de l'emplacement de la cellule en cours.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
SUBPICK(dimension, subset, vertical)
```

Argument	Description
dimension	Nom de la dimension contenant les sous-ensembles. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:dim. Pour un serveur distant, spécifiez nomserveur:dim.
subset	Nom du sous-ensemble dont vous voulez sélectionner des éléments.
vertical	Spécifiez TRUE pour insérer les noms d'élément verticalement, de haut en bas et en commençant par la cellule en cours. Spécifiez FALSE pour insérer les noms d'élément horizontalement, de la gauche vers la droite et en commençant par la cellule en cours.

### Exemple

```
=SUBPICK("local:Modèle", "Smith", TRUE)
```

Cet exemple insère les éléments sélectionnés du sous-ensemble Smith dans la feuille active. Les éléments sont disposés verticalement, de haut en bas et en commençant par la cellule en cours.

## T\_CLEAR

Cette fonction efface de la mémoire toutes les modifications ou ajouts apportés aux données de cube.

**Remarque :** T\_CLEAR ne vous invite pas à enregistrer sur disque les données de cube se trouvant dans la mémoire vive. Toutes les données non enregistrées sont effacées sans enregistrement sur disque. Par conséquent, si vous voulez enregistrer des données de cube actuellement en mémoire RAM, commencez par appeler la fonction T\_SAVE.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
T_CLEAR
```

### Arguments

Aucun.

## T\_CREATE

T\_CREATE crée un cube de huit dimensions au plus, nombre maximum de dimensions permis dans les versions antérieures de TM1.

**Remarque :** Si vous utilisez T\_CREATE pour créer un cube portant le nom d'un cube existant, TM1 remplace le cube existant et supprime toutes ses données.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.



## Syntaxe

```
T_CREATE(cube, d1, d2[, d3, d4, d5, d6, d7, d8])
```

Argument	Description
cube	Nom du cube à créer.  Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
d1...d8	Noms de huit dimensions existantes au maximum, dans l'ordre où vous voulez qu'elles soient stockées dans le cube. Vous devez spécifier au moins deux dimensions.

## Exemple

```
=T_CREATE("local:Ventes", "Région", "Produits", "Mois")
```

Cet exemple crée un cube nommé Ventes. Ce nouveau cube a trois dimensions se présentant dans l'ordre suivant : Région, Produits et Mois.

## T\_CREATE16

Cette fonction crée un cube de seize dimensions au plus.

**Remarque :** Si le premier argument de cette fonction est un nom de cube existant, TM1 remplace ce cube et supprime toutes ses données.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

## Syntaxe

```
T_CREATE16(cube, d1, d2[, d3, . . . , d16])
```

Argument	Description
cube	Nom du cube à créer.  Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
d1...d16	Noms de seize dimensions existantes au plus, dans l'ordre où vous voulez qu'elles soient stockées dans le cube. Vous devez spécifier au moins deux dimensions.

## Exemple

```
=T_CREATE("Ventes", "Région", "Produits", "Mois")
```

Cet exemple crée un cube nommé Ventes. Ce nouveau cube a trois dimensions se présentant dans l'ordre suivant : Région, Produits et Mois.

## T\_PICK

---

T\_PICK appelle une boîte de dialogue qui répertorie tous les cubes disponibles sur les serveurs TM1 locaux et distants. Le nom de cube que vous sélectionnez dans la boîte de dialogue devient la valeur de la fonction T\_PICK. Votre macro insère le nom du cube dans la première cellule de la feuille de calcul active.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
T_PICK
```

### Arguments

Aucun.

## T\_SAVE

---

Cette fonction enregistre sur disque toutes les données de cube actuellement en mémoire RAM. T\_SAVE ne peut être utilisée que pour enregistrer des données sur un serveur local. Elle ne fonctionne pas avec des serveurs distants. T\_SAVE ne vous invite pas à enregistrer les données pour les cubes individuels.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
T_SAVE
```

### Arguments

Aucun.

## TM1RECALC

---

Cette fonction force le programme à procéder à un nouveau calcul de toutes les feuilles ouvertes. Elle équivaut à appuyer sur F9 dans Excel. Une fonction de macros similaire, TM1RECALC1, force le programme à recalculer la feuille de calcul active.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
TM1RECALC
```

### Arguments

Aucun.

## TM1RECALC1

---

Cette fonction force le programme à procéder à un nouveau calcul de la feuille active. Elle équivaut à appuyer sur MAJ-F9 dans Excel. Une fonction de macros similaire, TM1RECALC, force le programme à recalculer toutes les feuilles de calcul ouvertes.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
TM1RECALC1
```

### Arguments

Aucun.

## VUSLICE

---

Cette fonction crée une feuille de tranche à partir de la vue de cube spécifiée. La tranche est insérée en commençant à la cellule supérieure de gauche (A1 ou R1C1) dans la feuille de calcul active.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
VUSLICE(cube, view)
```

Argument	Description
cube	Nom d'un cube existant. Utilisez un préfixe de nom de serveur pour indiquer l'emplacement du serveur. Pour le serveur local, spécifiez local:cube. Pour un serveur distant, spécifiez nom_serveur:cube.
view	Nom d'une vue associée au cube.

### Exemple

```
=VUSLICE("local:Ventés98","Trimestriel ")
```

Cet exemple copie les données de la vue Trimestrielle dans le cube Ventés98 de la feuille de calcul active.

## W\_DBSENABLE

---

Cette fonction active (ou désactive) le recalcul automatique des fonctions DBS dans une feuille de calcul.

Normalement, quand une fonction DBS est insérée dans une feuille de calcul, elle n'est exécutée que lorsque la feuille est recalculée à l'aide des touches F9 ou Maj+F9. Vous pouvez utiliser la fonction W\_DBSENABLE pour exécuter immédiatement des fonctions DBS à mesure qu'elles sont créées dans une feuille de calcul.

**Remarque :** Les fonctions DBS ne sont exécutables dans des modules VBA que si la fonction W\_DBSENABLE est définie à TRUE.

Cette fonction macro TM1 n'est valide que dans les macros Excel et les modules VBA.

### Syntaxe

```
=W_DBSENABLE(LogicalFlag)
```

Argument	Description
LogicalFlag	Avec TRUE, les fonctions DBS sont exécutées immédiatement lors de leur insertion ou appel depuis une feuille de calcul.  Avec FALSE, les fonctions DBS sont exécutées seulement quand la feuille de calcul est recalculée explicitement.

---

## Chapitre 4. Fonctions de feuilles de calcul

Les fonctions de feuille de calcul d'IBM Planning Analytics renvoient une valeur numérique ou une valeur de chaîne. Vous pouvez utiliser les fonctions de feuille de calcul TM1 n'importe où dans une feuille de calcul IBM Planning Analytics for Microsoft Excel ou TM1 Perspectives.

Pour accéder à ces fonctions dans Microsoft Excel, sélectionnez **Formules > Insérer une fonction**.

**Remarque :** Si vous utilisez Planning Analytics for Microsoft Excel, vous devez au préalable activer le module d'extension IBM Cognos Office Reporting TM1.

Si une fonction de feuille de calcul fait référence à un objet se trouvant sur un serveur distant, le nom de l'objet doit être précédé du nom du serveur et d'un deux-points. Par exemple, pour faire référence au cube SalesCube sur le serveur SData, utilisez SData:SalesCube. Vous devez être connecté au serveur auquel la fonction fait référence pour recevoir des valeurs exactes dans votre feuille de calcul. Si vous n'êtes pas connecté au serveur, les fonctions de feuille de calcul TM1 renvoient \*KEY\_ERR.

Les fonctions de feuille de calcul TM1 acceptent des chaînes, des valeurs ou des références de cellule comme arguments. Les chaînes doivent être mises entre guillemets. Les noms d'élément numérique doivent être mis entre des guillemets anglais doubles. Par exemple, ""14357"". Les références de cellules doivent faire référence à des arguments valides pour une fonction donnée. Vous pouvez utiliser les conventions standard pour les références de cellules relatives et absolues dans les fonctions de feuille de calcul.

En raison d'une limitation de Microsoft Excel, les fonctions de feuille de calcul ne peuvent pas contenir plus de 30 arguments. Lorsque vous créez une référence de cube, un argument doit être le nom du cube, ce qui laisse 29 arguments pour définir les dimensions du cube.

Si vous enregistrez dans Microsoft Excel une macro de feuille de calcul intégrant la fonctionnalité TM1, il est possible que la macro résultante contienne des fonctions de feuille de calcul TM1 non documentées. Cependant, il se peut que nous modifions ou cessions de prendre ces fonctions non documentées en charge dans les versions prochaines sans avis préalable.

Les fonctions de feuille de calcul ne peuvent **pas** être utilisées dans des règles TM1 ou des processus TurboIntegrator.

---

### DBR

La fonction DBR extrait une valeur d'un cube TM1 spécifié.

Lorsque tous les arguments d'éléments (e1, e2, etc.) de la fonction sont des éléments feuilles, la fonction DBR peut également être utilisée pour écrire les valeurs dans le cube spécifié, sous réserve que l'utilisateur dispose des privilèges d'accès appropriés pour le cube, les dimensions, les éléments et/ou les cellules concernés. Lorsque vous entrez une valeur dans une cellule contenant une telle fonction DBR, cette valeur est envoyée au serveur.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

#### Syntaxe

```
DBR(cube, e1, e2, [...en])
```

Argument	Description
cube	Nom du cube depuis lequel extraire la valeur.

Argument	Description
e1,...en	<p>Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.</p> <p>Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.</p> <p>Les noms d'élément numérique doivent être mis entre des guillemets anglais doubles. Par exemple, ""14357"".</p>

### Exemple

```
DBR("92act4d", "California", "3.5 Diskettes", "Net Sales", "January")
```

Dans cet exemple, 92act4d est le nom du cube et la fonction renvoie la valeur se trouvant à l'intersection de California, 3.5 Diskettes, Net Sales et January.

## DBRA

Cette fonction récupère la valeur d'un attribut d'élément spécifié. La valeur renvoyée peut être une chaîne ou une valeur numérique, selon le type de l'attribut.

La fonction DBRA peut aussi être utilisée pour écrire des valeurs d'attribut d'élément sur le serveur. Quand vous entrez une valeur, numérique ou chaîne, dans une cellule contenant une fonction DBRA, l'attribut d'élément correspondant est mis à jour sur le serveur.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DBRA(server:dimension, element, attribute)
```

Argument	Description
server:dimension	<p>Nom de dimension valide, précédé du nom de serveur approprié et du signe deux-points. Par exemple, "SData:Region" fait référence à la dimension Region sur le serveur SData.</p> <p>Si la dimension n'a pas de nom de serveur comme préfixe, la fonction DBRA tente de s'exécuter sur le serveur local.</p>
element	Elément de la dimension.
attribute	Attribut pour lequel vous voulez récupérer une valeur. Cet argument doit être un attribut valide de l'élément.

### Exemple

```
DBRA("SData:Model", "L Series 1.8L Sedan", "ManufactureCode")
```

Dans cet exemple, la fonction renvoie l'attribut ManufactureCode de l'élément L Series 1.8L Sedan de la dimension Model sur le serveur SData.

## DBRW

Cette fonction extrait une valeur d'un cube TM1 spécifié.

Lorsque tous les arguments d'éléments (e1, e2, etc.) de la fonction sont des éléments feuilles, la fonction DBRW peut également être utilisée pour écrire les valeurs dans le cube spécifié, sous réserve que l'utilisateur dispose des privilèges d'accès appropriés pour le cube, les dimensions, les éléments et/ou les cellules concernés.

DBRW opère comme DBR, avec une différence majeure ; DBRW réduit le trafic réseau et peut améliorer la performance dans les grands réseaux.

Dans les feuilles de calcul contenant un grand nombre de fonctions de feuille de calcul TM1, DBRW force TM1 à exécuter les fonctions en "lots" au lieu de les exécuter individuellement. Les fonctions DBR normales sont exécutées individuellement pendant le recalcul d'une feuille de calcul. Les fonctions DBRW forcent TM1 à réaliser deux passes sur la feuille de calcul. Pendant la première passe, toutes les valeurs qui ont changé dans les cellules contenant des fonctions DBRW sont envoyées au cube en un seul lot. Pendant la seconde passe, les valeurs du cube sont renvoyées à la feuille de calcul sous forme de lot unique. Par conséquent, la fonction recalcule deux fois lors de l'exécution de la fonction DBRW.

La mise en lots DBRW se produit lorsque la fonction est utilisée dans une cellule individuelle. Lorsque les fonctions DBRW sont utilisées dans le cadre de calculs complexes, la fonction agit au même titre qu'une fonction DBR et aucun gain de performance ne se fait sentir.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DBRW(cube, e1, e2[,...en])
```

Argument	Description
cube	Nom du cube depuis lequel extraire la valeur.
e1,...en	Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.  Les noms d'élément numérique doivent être mis entre des guillemets anglais doubles.

### Exemple

```
DBRW("92act4d", "California", "3.5 Diskettes", "NetSales", "January")
```

Dans cet exemple, la fonction renvoie la valeur se trouvant à l'intersection de California, 3.5 Diskettes, Net Sales et January dans le cube 92act4d.

## DBS

DBS envoie une valeur numérique à un cube TM1. Cette fonction ne peut pas envoyer de chaîne à un cube. Pour envoyer des chaînes, utilisez la fonction DBSS.

Quand vous générez une fonction DBS avec l'option **TM1 > Editer une formule**, la boîte de dialogue Editer une formule vous guide dans une série d'étapes permettant de générer chaque argument de la fonction dans l'ordre correct.

Si le cube n'existe pas ou si l'un des arguments n'est pas valide, la fonction renvoie KEY ERROR.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DBS(value, cube, e1, e2[,...en])
```

Argument	Description
value	Valeur envoyée.
cube	Cube auquel la valeur est envoyée.
e1, ...en	Noms des éléments définissant l'intersection de cube à laquelle la valeur est envoyée.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.  Les noms des éléments numériques doivent être mis entre guillemets.

### Exemple

Dans cet exemple, la fonction envoie la valeur 5342 dans le cube 92réel4d à l'intersection de Californie, Disquettes 3,5, Ventes nettes et Janvier.

```
DBS(5342,"92act4d","California","3.5 Diskettes", "NetSales", "January")
```

## DBSA

Cette fonction envoie une valeur à un attribut d'élément spécifié. La valeur envoyée peut être une chaîne ou une valeur numérique, selon le type de l'attribut.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DBSA(att_value, dimension, element, att_name)
```

Argument	Description
att_value	Valeur à envoyer.



Argument	Description
dimension	Nom de dimension valide. Ce nom doit être précédé du nom de serveur approprié et du signe deux-points. Par exemple, "SData:Region" fait référence à la dimension Region sur le serveur SData.  Si la dimension n'a pas de nom de serveur comme préfixe, la fonction DBSA tente de s'exécuter sur le serveur local.
element	Elément de la dimension.
att_name	Attribut auquel vous voulez envoyer une valeur. att_name doit être un attribut valide de l'élément spécifié par elem_name.

### Exemple

```
DBSA('LS-1.8-M7398', "SData:Model", "L Series 1.8LSedan", "Manufacture Code")
```

## DBSS

Cette fonction envoie une chaîne à un cube de n'importe quel nombre de dimensions. Cette fonction ne peut pas envoyer de valeur numérique à un cube. Utilisez la fonction DBS pour envoyer des valeurs numériques.

Quand vous générez une fonction DBSS avec l'option **TM1 > Editer une formule**, la boîte de dialogue Editer une formule vous guide dans une série d'étapes permettant de générer chaque argument de la fonction dans l'ordre correct.

Si le cube n'existe pas ou si l'un des arguments n'est pas valide, la fonction renvoie KEY ERROR.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DBSn(string, cube, e1, e2,...en)
```

Argument	Description
string	Chaîne envoyée.
cube	Cube auquel la chaîne est envoyée.
e1, ...en	Noms des éléments définissant l'intersection de cube à laquelle la chaîne est envoyée.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.

## Exemple

```
DBSS("Smith","Info","California","Last Name")
```

Dans cet exemple, la formule envoie la chaîne Smith au cube Info à l'intersection de California et de Last Name.

## DBSW

DBSW envoie une valeur numérique à un cube TM1. Cette fonction ne peut pas envoyer de chaîne à un cube. Pour envoyer des chaînes, utilisez la fonction DBSS.

Cette fonction opère de la même façon que la fonction DBS, à une différence importante près ; DBSW réduit le trafic réseau et peut améliorer la performance dans les grands réseaux.

Dans les feuilles de calcul contenant un grand nombre de références à des cubes, DBSW force TM1 à envoyer leurs valeurs par lots au lieu de les envoyer individuellement. Les fonctions DBS normales sont mises à jour individuellement pendant les recalculs. Les références DBSW forcent TM1 à envoyer toutes les valeurs qui ont été modifiées en un seul lot.

Dans de tels cas, vous pouvez utiliser une fonction DBS/DBR comme argument d'une autre fonction DBS en toute sécurité.

**Remarque :** Si vous utilisez VBA pour calculer une feuille de calcul contenant des fonctions DBSW, vous devez appeler la fonction macro `TM1` pour calculer la feuille de calcul. N'utilisez pas la méthode `VB Calculate` pour calculer une feuille de calcul contenant des fonctions DBSW ; ceci entraînerait l'exécution individuelle de chaque fonction DBSW, ce qui annulerait l'avantage que présente la fonction et résulterait en une baisse de performance.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

## Syntaxe

```
DBSW(value, cube, e1, e2[,...en])
```

Argument	Description
value	Valeur envoyée.
cube	Cube auquel la valeur est envoyée.
e1, ...en	Noms des éléments définissant l'intersection de cube à laquelle la valeur est envoyée.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.  Les noms des éléments numériques doivent être mis entre guillemets.

## Exemple

```
DBSW(5342,"92act4d","California","3.5 Diskettes", "NetSales", "January")
```

## DFRST

---

DFRST renvoie le premier élément d'une dimension spécifiée.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DFRST(server_name:dimension)
```

Argument	Description
dimension	Nom de dimension valide.

### Exemple

```
DFRST("planning_sample:Location")
```

Si la dimension Location contient, dans cet ordre, les éléments California, Oregon et Washington, l'exemple renvoie California.

## DIMIX

---

DIMIX renvoie le numéro d'index d'un élément dans une dimension.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DIMIX(server_name:dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension.
	Si l'élément n'est pas membre de la dimension spécifiée, la fonction renvoie la valeur 0. Cet argument peut également être le nom d'un alias d'un élément de dimension.

### Exemple

```
DIMIX("planning_sample: Location", "Washington")
```

Si la dimension Location contient, dans cet ordre, les éléments California, Oregon et Washington, l'exemple renvoie la valeur 3, Washington étant le troisième élément de la dimension.

## DIMNM

DIMNM renvoie l'élément d'une dimension correspondant à l'argument Index. Si vous ajoutez le paramètre facultatif Alias à cette fonction, elle renvoie l'alias de l'élément sélectionné.

Quand vous double-cliquez sur une cellule contenant une fonction DIMNM, la boîte de dialogue Dimension s'affiche. Vous pouvez alors sélectionner un nouvel élément à mettre dans votre feuille de calcul. La fonction DIMNM met automatiquement à jour l'argument Index pour refléter le nouvel élément.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DIMNM(server_name:Dimension, Index, [Alias])
```

Argument	Description
Dimension	Nom de dimension valide.
Index	Valeur inférieure ou égale au nombre d'éléments dans la dimension.
Alias	Nom d'un alias existant pour la dimension. Cet argument est facultatif. S'il est utilisé, la fonction renvoie l'alias pour l'élément spécifié.

## DIMSIZ

DIMSIZ renvoie le nombre d'éléments d'une dimension spécifiée.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DIMSIZ(dimension)
```

Argument	Description
dimension	Nom de dimension valide.

### Exemple

```
DIMSIZ("Accounts")
```

Si la dimension Accounts contient 19 éléments, l'exemple renvoie la valeur 19.

## DNEXT

DNEXT renvoie le nom de l'élément qui suit celui spécifié comme argument de la fonction.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DNEXT(server:dimension, element)
```

Argument	Description
serveur:dimension	Nom de dimension valide, précédé du nom de serveur approprié et du signe deux-points. Par exemple, "SData:Region" fait référence à la dimension Region sur le serveur SData.  Si la dimension n'est pas précédée d'un nom de serveur, la fonction DBRA tente de s'exécuter sur le serveur local.
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
DNEXT("Production:Location", "Oregon")
```

Si la dimension Location sur le serveur Production contient, dans cet ordre, les éléments California, Oregon et Washington, l'exemple renvoie Washington.

## DNLEV

DNLEV renvoie le nombre de niveaux de hiérarchie d'une dimension.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

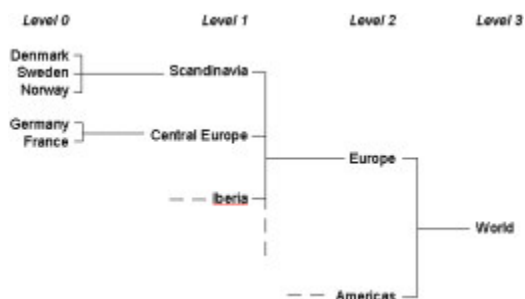
```
DNLEV(dimension)
```

Argument	Description
dimension	Nom de dimension valide.

### Exemple

```
DNLEV("Region")
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3).



Il y a quatre niveaux de hiérarchie dans la dimension Region (0, 1, 2 et 3). Par conséquent, l'exemple renvoie la valeur 4.

## DTYPE

DTYPE renvoie des informations sur le type d'élément de l'élément spécifié. Elle renvoie "N" si l'élément est numérique, "S" s'il s'agit d'une chaîne.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
DTYPE(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
TYPE("Region", "Europe")
```

L'élément Europe de la dimension Region est un élément consolidé et l'exemple renvoie donc "C".

## ELCOMP

ELCOMP renvoie le nom d'un enfant d'un élément consolidé d'une dimension spécifiée. Si l'argument de l'élément n'est pas un élément consolidé, la fonction renvoie la valeur 0.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELCOMP(dimension, element, index)
```

Argument	Description
dimension	Nom de dimension valide.
élément	Nom d'un élément consolidé dans la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.
index	Valeur positive inférieure ou égale au nombre total d'enfants dans l'élément spécifié.

### Exemple

```
ELCOMP("Region", "Central Europe", 2)
```

Dans la dimension Region, l'élément consolidé Central Europe est une consolidation des enfants Germany et France. Par conséquent, l'exemple renvoie France.

## ELCOMPN

ELCOMPN renvoie le nombre de composants d'un élément spécifié. Si l'argument de l'élément n'est pas un élément consolidé, la fonction renvoie la valeur 0.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELCOMPN(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément consolidé dans la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

Dans la dimension Region, l'élément Scandinavie est une consolidation de trois éléments. L'exemple renvoie 3.

## ELISCOMP

ELISCOMP détermine si element1 est un enfant d'element2 dans la dimension spécifiée. La fonction renvoie TRUE si élément1 est un enfant d'élément2, sinon elle renvoie FALSE.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELISCOMP(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.
element2	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELISCOMP("Region", "Germany", "Central Europe")
```

Dans la dimension Région, l'élément Europe occidentale est une consolidation des deux éléments Allemagne et France. L'exemple renvoie donc TRUE.

Notez que cette fonction ne renvoie TRUE que pour les enfants immédiats. Dans l'exemple ci-dessus, Germany est un enfant de Central Europe. De plus, Europe occidentale est un enfant de Europe. Cependant, puisque la fonction ne renvoie TRUE que pour les enfants immédiats, l'exemple ci-dessous renvoie FALSE :

```
ELISCOMP("Region", "Germany", "Europe")
```

## ELISPAR

ELISPAR détermine si element1 est un parent d'element2 dans la dimension spécifiée. La fonction renvoie TRUE si element1 est un parent d'element2, sinon elle renvoie FALSE.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELISPAR(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.
element2	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELISPAR("Region", "Central Europe", "Germany")
```

Dans la dimension Region, l'élément consolidé Central Europe est le parent de Germany et France. Par conséquent, l'exemple renvoie TRUE.

Notez que cette fonction ne renvoie TRUE que pour les parents immédiats. Dans l'exemple ci-dessus, Europe est un parent de Europe occidentale. De plus, Europe occidentale est un parent de Allemagne. Toutefois, Europe n'étant pas un parent immédiat de Germany, l'exemple ci-dessous renvoie FALSE.

```
ELISPAR("Region", "Europe", "Germany")
```

## ELLEV

ELLEV renvoie le niveau d'un élément d'une dimension.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELLEV(dimension, element)
```

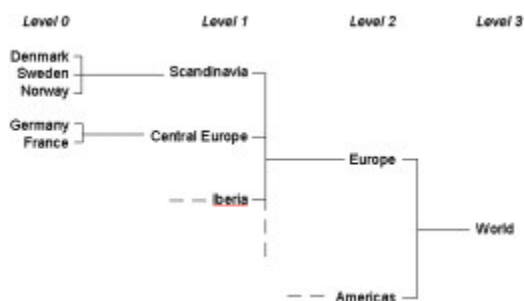


Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELLEV("Region", "Europe")
```

Dans la dimension Region, les différents pays (Niveau 0) composent les régions (Niveau 1). Les régions composent elles-mêmes les super-régions (Niveau 2) qui composent enfin le monde (Niveau 3).



L'exemple renvoie 2, Europe étant un élément du niveau 2.

## ELPAR

ELPAR renvoie le parent d'un élément d'une dimension spécifiée.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELPAR(dimension, element, index)
```

Argument	Description
dimension	Nom de dimension valide.
élément	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.
index	Valeur positive inférieure ou égale au nombre total d'éléments consolidés (parents) utilisant l'argument d'élément comme enfant.

### Exemple

```
ELPAR("Model", "Wagon 4WD", 2)
```

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. Dans la structure de la dimension Model, Total Wagons est défini en premier et Total 4WD est défini en second.

L'exemple renvoie Total 4WD, deuxième instance d'un parent de Wagon 4WD dans la dimension Model.

## ELPARN

ELPARN renvoie le nombre de parents d'un élément d'une dimension spécifiée.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELPARN(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément de la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELPARN("Model", "Wagon 4WD")
```

Dans la dimension Model, l'élément Wagon 4WD est un enfant de Total Wagons et de Total 4WD. Par conséquent, Total Wagons et Total 4WD sont des parents de Wagon 4WD. La fonction renvoie 2.

## ELSLEN

ELSLEN renvoie la longueur d'un élément chaîne d'une dimension. Si l'élément spécifié ne fait pas partie de la dimension spécifiée ou n'est pas une chaîne, la fonction renvoie 0.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELSLEN(dimension, element)
```

Argument	Description
dimension	Nom de dimension valide.
element	Nom d'un élément chaîne dans la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELSLEN("Region", "Washington")
```

L'élément Washington est un élément chaîne de dix caractères de longueur. L'exemple renvoie la valeur 10.

## ELWEIGHT

ELWEIGHT renvoie la pondération d'un enfant d'un élément consolidé.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
ELWEIGHT(dimension, element1, element2)
```

Argument	Description
dimension	Nom de dimension valide.
element1	Nom d'un élément consolidé dans la dimension. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.
element2	Nom d'un enfant de l'élément consolidé. Cet argument peut aussi être le nom d'un alias pour un élément de la dimension.

### Exemple

```
ELWEIGHT("Account1", "Gross margin", "Variable costs")
```

Comme le montre la figure suivante, l'élément Variable costs, qui est un enfant de Gross margin, a une pondération de -1.

Children of 'Gross margin'		
Name	Type	Weight
<a href="#">Sales</a>	Simple	1
<a href="#">Variable Costs</a>	Simple	-1

L'exemple renvoie -1.

## SUBNM

Cette fonction renvoie l'élément d'un sous-ensemble de dimensions correspondant à l'argument IndexOrName. Si vous ajoutez le paramètre facultatif Alias à cette fonction, elle renvoie l'alias de l'élément sélectionné.

Quand vous double-cliquez sur une cellule contenant une fonction SUBNUM, l'Editeur de sous-ensemble s'ouvre. Vous pouvez alors sélectionner un nouvel élément à mettre dans votre feuille de calcul.

L'élément sélectionné devient la valeur renvoyée de la fonction SUBNM et celle-ci met automatiquement à jour l'argument IndexOuNom pour refléter le nouvel élément.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
SUBNM(Dimension, Subset, IndexOrName, [Alias])
```

Argument	Description
Dimension	Nom de dimension valide.
Subset	Nom d'un sous-ensemble de la dimension.
IndexOrName	Index du sous-ensemble ou nom d'un élément du sous-ensemble.  Dans le cas d'un index, un entier positif inférieur ou égal au nombre total d'éléments du sous-ensemble spécifié. Dans le cas d'un nom, une chaîne représentant le nom d'un élément du sous-ensemble.
Alias	Nom d'un alias existant pour le sous-ensemble. Cet argument est facultatif. S'il est utilisé, l'alias spécifique est appliqué quand l'Editeur de sous-ensemble s'ouvre et la fonction renvoie l'alias de l'élément sélectionné.

### Exemple

```
SUBNM("Region", "Top Producers", 2)
```

Le sous-ensemble Principaux producteurs de la dimension Region contient, en ordre, les éléments United States, Germany, Great Britain et Mexico. Puisque l'argument index pointe au second élément du sous-ensemble, l'exemple renvoie Germany.

```
SUBNM("Region", "Top Producers", "Germany", "Deutsch")
```

Cet exemple renvoie l'alias Allemand pour l'élément Germany (Deutschland) depuis le sous-ensemble Top Producers de la dimension Region.

## SUBSIZ

SUBSIZ renvoie le nombre éléments d'un sous-ensemble de dimension.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
SUBSIZ(dimension, subset)
```

Argument	Description
dimension	Nom de dimension valide.
subset	Nom d'un sous-ensemble de la dimension.

### Exemple

```
SUBSIZ("Region", "Top Producers")
```

Le sous-ensemble Principaux producteurs de la dimension Region contient, en ordre, quatre éléments United States, Germany, Great Britain et Mexico.

L'exemple renvoie 4.

## TABDIM

TABDIM renvoie le nom de dimension correspondant à un argument index donné.

La fonction renvoie toujours une dimension d'après l'ordre initial des dimensions dans le cube spécifié, même si l'ordre des dimensions dans le cube a été modifié à l'aide de l'Optimiseur de cube TM1.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TABDIM(cube, index)
```

Argument	Description
cube	Nom de cube valide.
index	Valeur positive inférieure ou égale au nombre total de dimensions du cube.

### Exemple

```
TABDIM("98sales",3)
```

Le cube 98sales compte cinq dimensions : account1, actvsbud, model, month et region. L'exemple renvoie model, la troisième dimension de 98sales.

## TM1ELLIST

La fonction TM1ELLIST renvoie un vecteur de valeurs de tableau vers le bas. Elle est utile car à l'aide d'une formule unique, vous pouvez obtenir un ensemble de valeurs d'élément à partir d'un modèle TM1.

### Remarque :

- TM1ELLIST ne remplace pas ou n'est pas inséré dans un espace autre qu'une formule tel qu'un Rapport dynamique. Il est de la responsabilité du concepteur du classeur de consommer correctement la réponse comportant plusieurs valeurs.
- TM1ELLIST renvoie un tableau de valeurs, mais seul le premier élément s'affiche si la fonction est placée dans un magasin de valeurs singulières.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1ELLIST(ServerDimension, [SetName], [ElementList],  
[AliasOverride], [ExpandAbove], [MDXOverride], [IndentRate], [IndentCharacter])
```

Argument	Description	Obligatoire/Facultatif
ServerDimension	Dimension spécifiée au format serveur:dimension.	Requis
SetName	Ensemble nommé. Si cet argument est vide, tous les éléments de la dimension sont utilisés.	Facultatif

Argument	Description	Obligatoire/Facultatif
Elementlist	<p>Tableau de valeurs qui indique une liste d'éléments constituant un ensemble. Par exemple, ElementList peut faire référence à une plage de cellules.</p> <p>Quand cet argument est fourni, l'ensemble nommé spécifié par l'argument SetName est ignoré.</p> <p>Quand cet argument est vide, les éléments de l'ensemble spécifié par l'argument SetName sont utilisés.</p>	Facultatif
AliasOverride	<p>Chaîne définissant l'alias utilisé pour l'ensemble.</p> <p>Quand cet argument est fourni, il supprime la propriété d'alias par défaut définie par le sous-ensemble spécifié par l'argument SetName.</p> <p>Si cet argument est vide, les alias provenant de l'ensemble spécifié par l'argument SubsetName sont utilisés.</p>	Facultatif
ExpandAbove	<p>Indicateur booléen activant ou désactivant la propriété de l'ensemble ExpandAbove.</p> <p>Quand cet argument est fourni, il supprime la propriété par défaut ExpandAbove définie par le sous-ensemble spécifié par l'argument Set.</p> <p>Si cette valeur d'argument est 1, les membres consolidés se développent vers le haut lors de l'accès au détail.</p> <p>Si la valeur d'argument est 0, les membres consolidés se développent vers le bas lors de l'accès au détail. Si cet argument est vide, la propriété ExpandAbove provenant du sous-ensemble spécifié par l'argument Subset est utilisée.</p>	Facultatif
MDXOverride	<p>Instruction MDX qui s'applique au sous-ensemble spécifié par l'argument SubsetName/ElementList.</p> <p>Quand cet argument est fourni, il supprime le filtre MDX par défaut défini par le sous-ensemble spécifié par l'argument SetName.</p> <p>Si cet argument est vide ou omis, les membres de l'ensemble spécifié par l'argument SetName sont utilisés.</p>	Facultatif

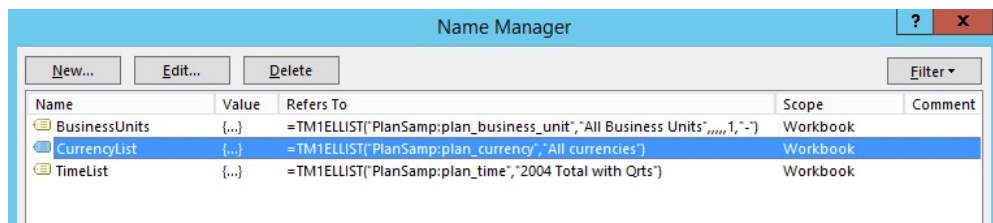
Argument	Description	Obligatoire/Facultatif
IndentRate	Entier indiquant combien d'indentations sont appliquées à chaque niveau lors du passage au niveau inférieur d'un membre consolidé.  Si la valeur de cet argument est 0, il n'y a aucune indentation automatique effectuée. IndentRate est relatif au niveau d'ensemble des éléments de l'ensemble.  Si cet argument est vide ou omis, une indentation est appliquée à chaque niveau lors du passage au niveau inférieur d'un membre consolidé.	Facultatif
IndentCharacter	IndentChar définit le symbole utilisé pour l'indentation dans la chaîne ; la valeur par défaut est le demi-cadratin (symbole d'espace normal).	Facultatif

### Exemple

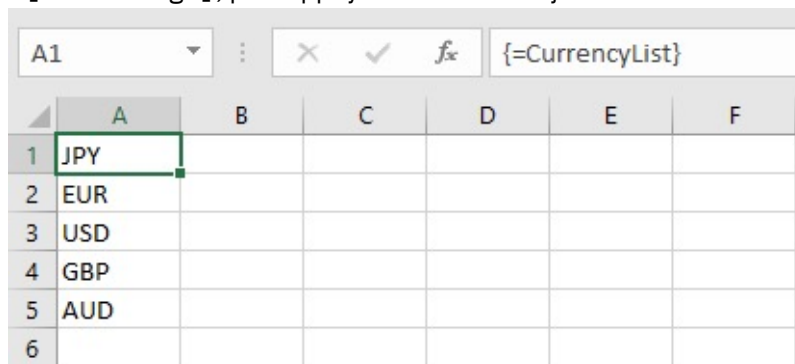
```
TM1ELLIST("PlanSamp:plan_currency","All currencies")
```

TM1ELLIST renvoie un tableau d'éléments basé sur des arguments donnés.

Pour extraire toutes les valeurs renvoyées, une plage nommée peut être créée dans Excel, faisant référence à la formule. Pour cela, créez une plage nommée et entrez la formule TM1ELLIST dans la colonne **Refers To**.



Sélectionnez le nombre voulu de cellules (basé sur la taille de tableau renvoyée) dans Excel, tapez `=[namedrange]`, puis appuyez sur CTRL+Maj+Entrée.



La fonction Excel INDEX peut être utilisée pour extraire un seul élément pour la plage.

	A	B	C	D	E	F	G
1	JPY						
2	EUR	EUR					
3	USD						
4	GBP						
5	AUD						
6							

## TM1GLOBALSANDBOX

TM1GLOBALSANDBOX renvoie le bac à sable actif global actuel de l'utilisateur.

**Remarque :** Cette fonction est valide uniquement dans les pages Web Planning Analytics for Microsoft Excel et Planning Analytics. Elle n'est pas prise en charge dans IBM TM1 Perspectives.

### Syntaxe

TM1GLOBALSANDBOX (SERVER)

Argument	Description	Obligatoire/Facultatif
Serveur	Nom du serveur TM1.	Requis

### Exemple

TM1GLOBALSANDBOX("Planning Sample")

## TM1INFO

TM1INFO renvoie des informations sur le client et la version TM1 actuels.

**Remarque :** Cette fonction est valide uniquement dans les pages Web Planning Analytics for Microsoft Excel et Planning Analytics. Elle n'est pas prise en charge dans IBM TM1 Perspectives.

### Syntaxe

TM1INFO("Property Name")



Argument	Description	Obligatoire/Facultatif
Nom de la propriété	<p>Le nom de la propriété peut correspondre à l'un des noms suivants :</p> <p><b>clientversion</b> Renvoie le numéro de version complet du client. Par exemple, 10.2.10000</p> <p><b>clientversionmajor</b> Renvoie le numéro de version majeur du client.</p> <p><b>clientversionminor</b> Renvoie le numéro de version mineur du client.</p> <p><b>clientversionpatch</b> Renvoie le groupe de correctifs et le numéro de correctif logiciel.</p> <p><b>client</b> Renvoie le nom du client. Par exemple, cor ou websheet.</p>	Requis

### Exemple

```
TM1INFO("clientversion")
```

## TM1PRIMARYDBNAME

TM1PRIMARYDBNAME renvoie le nom du serveur TM1 principal par l'intermédiaire duquel l'utilisateur est authentifié, même si l'utilisateur est connecté implicitement à plusieurs serveurs TM1. Cette fonction n'accepte aucun argument.

**Remarque :** Cette fonction est valide uniquement dans les pages Web Planning Analytics for Microsoft Excel et Planning Analytics. Elle n'est pas prise en charge dans IBM TM1 Perspectives.

### Syntaxe

```
TM1PRIMARYDBNAME()
```

## TM1RptElIsConsolidated

Cette fonction renvoie une valeur booléenne pour indiquer si un élément d'un Formulaire actif est consolidé. Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1RptElIsConsolidated(RptRowFormula, Element)
```

Argument	Description
RptRowFormula	Référence absolue à une cellule contenant une formule TM1RptRow.

Argument	Description
Element	Référence relative à une cellule contenant un élément de la formule TM1RptRow.

## TM1RptEIsExpanded

Cette fonction renvoie une valeur booléenne pour indiquer si un élément est développé dans un sous-ensemble de ligne d'un formulaire actif Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1RptEIsExpanded(RptRowFormula, Element)
```

Argument	Description
RptRowFormula	Référence absolue à une cellule contenant une formule TM1RptRow.
Element	Référence relative à une cellule contenant un élément de la formule TM1RptRow.

## TM1RptE1Lev

Cette fonction renvoie une valeur entière pour un niveau d'élément relatif à la racine dans le sous-ensemble. Cette fonction de feuille de calcul permet de créer des formulaires actifs. Cette fonction est distincte de la fonction de feuille de calcul E1Lev.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1RptE1Lev(RptRowFormula, Element)
```

Argument	Description
RptRowFormula	Référence absolue à une cellule contenant une formule TM1RptRow.
Element	Référence relative à une cellule contenant un élément de la formule TM1RptRow.

## TM1RptFilter

Cette fonction définit le filtre appliqué à une dimension en colonne d'un formulaire actif. Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

## Syntaxe

TM1RptFilter(ReportView, Tuple, FilterFunction, FilterValue, SortOrder)

Argument	Description
ReportView	Une référence de cellule contenant une formule TM1RptView. Ce filtre s'applique à la vue spécifiée par la formule TM1RptView.
Tuple	Une chaîne de tuple spécifiant l'élément de la dimension en colonne auquel le filtre s'applique. Par exemple, [month].[Feb].
FilterFunction	Un des noms de fonction de filtre suivants : TOPCOUNT BOTTOMCOUNT TOPPERCENT BOTTOMPERCENT TOPSUM BOTTOMSUM
FilterValue	Valeur de filtre.
SortOrder	Un des deux ordres de tri suivants : asc (croissant) desc (décroissant)

## Exemple

```
=TM1RptFilter($B$4, "[month].[Jan]", "TOPCOUNT", 5, "asc")
```

## TM1RptRow

Cette fonction détermine la définition de ligne maîtresse du formulaire actif. La définition de la ligne maîtresse régit le comportement de toutes les lignes du Formulaire actif. Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

## Syntaxe

TM1RptRow(ReportView, Dimension, Subset, SubsetElements, Alias, ExpandAbove, MDXStatement, Indentations, ConsolidationDrilling)

Argument	Description
ReportView	Référence à une cellule contenant une formule TM1RptView.

Argument	Description
Dimension	Dimension spécifiée au format nom_serveur_tm1:nom_dimension.
Subset	Sous-ensemble nommé. Si cet argument est vide, tous les éléments de la dimension sont utilisés
SubsetElements	<p>Référence à une plage de cellules qui indique une liste d'éléments constituant un sous-ensemble.</p> <p>Quand cet argument est fourni, le sous-ensemble nommé spécifié par l'argument Subset est ignoré.</p> <p>Quand cet argument est vide, les éléments du sous-ensemble spécifié par l'argument Subset sont utilisés.</p>
Alias	<p>Chaîne définissant l'alias utilisé pour le sous-ensemble.</p> <p>Quand cet argument est fourni, il supprime la propriété d'alias par défaut définie par le sous-ensemble spécifié par l'argument Sous-ensemble.</p> <p>Si cet argument est vide, les alias provenant du sous-ensemble spécifié par l'argument Sous-ensemble sont utilisés.</p>
ExpandAbove	<p>Indicateur booléen activant ou désactivant la propriété de sous-ensemble ExpandAbove. Quand cet argument est fourni, il supprime la propriété par défaut ExpandAbove définie par le sous-ensemble spécifié par l'argument Subset.</p> <p>Si cette valeur d'argument est 1, les éléments consolidés se développent vers le haut lors de l'accès au détail.</p> <p>Si cette valeur d'argument est 0, les éléments consolidés se développent vers le bas lors de l'accès au détail.</p> <p>Si cet argument est vide, la propriété ExpandAbove provenant du sous-ensemble spécifié par l'argument Subset est utilisée.</p>
MDXStatement	<p>Instruction MDX qui s'applique au sous-ensemble spécifié par l'argument Subset.</p> <p>Quand cet argument est fourni, il supprime le filtre MDX par défaut défini par le sous-ensemble spécifié par l'argument Subset.</p> <p>Quand cet argument est vide ou omis, les éléments du sous-ensemble spécifié par l'argument Subset sont utilisés.</p>

Argument	Description
Indentations	Entier indiquant combien d'indentations sont appliquées à chaque niveau lors du passage au niveau inférieur d'un élément consolidé. Si la valeur de cet argument est 0, il n'y a aucune indentation automatique effectuée.  Cet argument est facultatif. Si cette valeur manque, une indentation est appliquée à chaque niveau lors du passage au niveau inférieur d'un élément consolidé.
ConsolidationDrilling	Indicateur booléen activant ou désactivant l'accès au détail des éléments consolidés.  Quand la valeur de cet argument est 1, les utilisateurs peuvent passer au niveau inférieur des éléments consolidés du formulaire actif.  Quand la valeur de cet argument est 0, les utilisateurs ne peuvent pas passer au niveau inférieur des éléments consolidés du formulaire actif.  Cet argument est facultatif. Si l'argument manque, le comportement par défaut est de permettre l'accès au détail des éléments consolidés.

### Exemple

```
=TM1RptRow($B$9,"sdata:region","", '{AR}01'!$B$17:$B$18,"",1,"",5, 0)
```

## TM1RptTitle

Cette fonction définit la dimension du titre du formulaire actif. Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1RptTitle(Dimension,Element)
```

Argument	Description
Dimension	Dimension spécifiée au format nom_tm1:nom_dimension.
Element	Référence à une cellule contenant une fonction SUBNM qui renvoie un nom d'élément.

### Exemple

```
TM1RptTitle("SData:model", $C$7)
```

## TM1RptView

TM1RptView définit la vue affichée dans un Formulaire actif. Cette fonction de feuille de calcul permet de créer des formulaires actifs.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1RptView(ViewID,ZeroSuppression,TM1RptTitle,...)
```

Argument	Description
IDVue	Nom de la vue au format nom_tm1:nom_cube:id_unique.
ZeroSuppression	Un indicateur booléen activant ou désactivant la propriété de suppression des valeurs nulles pour la vue. 1 = activé, 0 = désactivé
TM1RptTitle	Pour chaque dimension en titre du formulaire actif, inclure une fonction TM1RptTitle en tant qu'argument de TM1RptView.
FormatRange	La plage de formatage pour le Formulaire actif. Lorsque vous créez un Formulaire actif, une plage nommée appelée TM1RPTFMTRNG est créée pour inclure toutes les cellules de la plage de formatage. Vous pouvez utiliser cette plage nommée comme argument.
ColonneIDs	La colonne contenant les ID de format du Formulaire actif. Lorsque vous créez un Formulaire actif, une plage nommée appelée TM1RPTFMTIDCOL est créée pour inclure toutes les cellules de la plage de formatage. Vous pouvez utiliser cette plage nommée comme argument.

### Exemple

```
=TM1RPTVIEW("SData:SalesCube:6", 0,  
TM1RPTTITLE("SData:actvsbud", $C$6),  
TM1RPTTITLE("SData:model", $C$7),  
TM1RPTTITLE("SData:account1", $C$8),  
TM1RPTFMTRNG,  
TM1RPTFMTIDCOL)
```

## TM1User

TM1User renvoie le nom d'utilisateur de l'utilisateur TM1 actuel.

Si l'utilisateur TM1 actuel n'est pas connecté à un serveur, ou si le serveur spécifié n'est pas en marche, TM1User renvoie une chaîne vide.

Si TM1User est exécutée face à un serveur configuré pour authentification CAM, la fonction renvoie le nom interne de l'utilisateur/CAMID et non pas son nom d'affichage.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
TM1User("ServerName")
```

Argument	Description
ServerName	Nom du serveur auquel l'utilisateur TM1 est connecté.

### Exemple

```
TM1User("SData")
```

Si un utilisateur nommé BrianT est connecté au serveur SData et que cet utilisateur exécute la fonction TM1User, l'exemple ci-dessus renvoie la fonction BrianT.

## VIEW

Cette fonction crée une vue optimisée du cube spécifié par l'argument de cube.

Une fonction VIEW unique est créée quand vous faites une tranche de vue à partir d'une navigation de cube.

Toutes les formules DBR et DBRW faisant référence à la fonction VIEW peuvent alors accéder à cette vue unique. De cette façon, les résultats sont renvoyés beaucoup plus rapidement.

Plusieurs fonctions VIEW peuvent résider dans la même feuille de calcul si vous avez des blocs de formules faisant référence à des vues et/ou des cubes TM1 différents.

Cette fonction de feuille de calcul n'est valide que dans les feuilles de calcul.

### Syntaxe

```
VIEW(cube, e1,e2[,...en])
```

Argument	Description
cube	Nom du cube depuis lequel extraire les données.
e1,...en	Soit des éléments spécifiques de la tranche à utiliser comme titres, soit la chaîne "!". La chaîne "!" indique que la dimension correspondante est une ligne ou une colonne de la vue. Ces arguments peuvent aussi être les noms d'alias des éléments de dimension.

### Exemple

```
VIEW("93sales", $B$2, $B$3, $B$4, "!", "!")
```





---

## Chapitre 5. fonctions TurboIntegrator

TM1 TurboIntegrator vous permet de manipuler des données et des métadonnées TM1 lors de la définition d'un processus.

Ceci est accompli par le biais de fonctions sur les sous-onglets Prologue, Métadonnées, Données et Epilogue de l'onglet Expert dans la fenêtre TurboIntegrator. Ces sous-onglets contiennent des instructions générées d'après les paramètres et les options sélectionnés lors de la définition d'un processus TurboIntegrator. Toutes les fonctions que vous créez doivent apparaître après les instructions générées. Pour plus de détails sur la création de processus avec TurboIntegrator, reportez-vous à la documentation de *TM1 TurboIntegrator*.

Les fonctions TurboIntegrator décrites dans cette section sont organisées par catégories.

Aucune interface n'est disponible pour aider à créer des fonctions TurboIntegrator. Vous devez saisir ces fonctions manuellement dans le sous-onglet approprié de l'onglet Expert. Les arguments chaîne des fonctions TurboIntegrator doivent être placés entre guillemets simples. Un point-virgule (;) doit indiquer la fin de chaque fonction dans la fenêtre TurboIntegrator.

En plus des fonctions TurboIntegrator, vous pouvez incorporer toutes les fonctions de règle TM1 standard dans les processus de définition, à l'exception de la fonction STET.

Chacun des arguments des fonctions TurboIntegrator est limité à 256 octets. Une fonction TurboIntegrator accepte plusieurs arguments, chacun étant limité à 256 octets.

---

### Mots réservés de TurboIntegrator

Pour empêcher les erreurs dans vos scripts TurboIntegrator, évitez de créer des variables ayant des noms identiques aux mots répertoriés dans les catégories suivantes.

Il y a quatre catégories de mots réservés dans TurboIntegrator :

- Noms de fonction de règles : pour la liste complète de tous les noms de fonction de règles, reportez-vous à la rubrique [Chapitre 2, «Fonctions de règles»](#), à la page 103.
- Noms de fonction de TurboIntegrator : pour la liste complète de tous les noms de fonction de TurboIntegrator, reportez-vous à la rubrique [Chapitre 5, «fonctions TurboIntegrator»](#), à la page 217.
- Noms de variable locale implicites : pour une liste complète de tous les noms de variable locale implicites de TurboIntegrator, reportez-vous à la rubrique [«Variables locales TurboIntegrator»](#), à la page 419.
- Mots clés de TurboIntegrator : ces mots clés sont réservés et ne doivent pas être utilisés comme variables dans vos scripts :
  - break
  - else
  - elseif
  - end
  - endif
  - if
  - while

---

### Fonctions TurboIntegrator pour ASCII et texte

Ces fonctions concernent ASCII et le texte.

## ASCIIDelete

Cette fonction supprime un fichier ASCII.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ASCIIDelete(FileName);
```

Argument	Description
FileName	Nom du fichier ASCII à supprimer. Si aucun chemin complet n'est spécifié, TM1 recherche le fichier dans le répertoire de données du serveur.

### Exemple

Dans cet exemple, le fichier ASCII nommé 2002Q1Results.cma est supprimé du répertoire C:\exported\_data.

```
ASCIIDelete('C:\exported_data\2002Q1Results.cma');
```

## ASCIIOutput

Cette fonction écrit un enregistrement à virgules de séparation dans un fichier ASCII.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Le fichier ASCII est ouvert lors de l'écriture du premier enregistrement et fermé à la fin de la procédure TurboIntegrator (Prologue, Métadonnées, Données, Epilogue) contenant la fonction ASCIIOutput.

Tous les enregistrements en sortie générés par ASCIIOutput sont limités à 8000 octets. Si un enregistrement en sortie dépasse 8000 octets, cet enregistrement est tronqué et un avertissement est écrit dans le fichier TM1ProcessError.log.

Quand ASCIIOutput rencontre un argument chaîne qui pousse l'enregistrement en sortie au-delà de la limite de 8000 octets, il ignore cet argument et tous les arguments qui suivent. Par exemple, s'il y a 10 arguments chaînes et que la sortie pour les sept premiers arguments atteint 7950 octets alors que la sortie pour le huitième argument est de 51 octets, seule la sortie des sept premiers arguments est écrite dans l'enregistrement. S'il y a dix arguments chaînes et que le premier argument dépasse 8000 octets, aucune sortie n'est écrite dans l'enregistrement.

Si vous utilisez la fonction ASCIIOutput pour écrire dans le même fichier dans plusieurs procédures (onglets) d'un processus TurboIntegrator, ce fichier est remplacé chaque fois qu'il est ouvert pour une nouvelle procédure.

La fonction ASCIIOutput génère une erreur mineure si une erreur se produit lors de l'écriture du fichier ASCII. En outre, la fonction renvoie une valeur de l'exécution : 1 si la fonction écrit avec succès le fichier texte et 0 en cas d'échec.

**Remarque :** L'erreur est générée et la valeur est renvoyée uniquement lorsque ASCIIOutput écrit sur un disque autre que celui sur lequel le serveur s'exécute. Par exemple, si le serveur tourne sur le lecteur C:, que ASCIIOutput écrit sur le lecteur F: et que F: vient à manquer d'espace, l'erreur est piégée et le serveur reste actif. Si le serveur s'exécute sur l'unité C: alors que ASCIIOutput écrit également sur cette unité et que l'espace vient à manquer, le serveur sera arrêté (comme prévu).

**Remarque :** La possibilité d'exécuter ou non la fonction ASCIIOutput lorsque la source de données est une vue de cube est déterminée par l'affectation de la fonction **Exporter comme données texte**, qui est définie par groupe d'utilisateurs. Si un utilisateur est membre d'un groupe qui n'est pas autorisé à exporter les données sous forme de texte, toute tentative par cet utilisateur d'exécuter ASCIIOutput

entraîne l'arrêt du processus avec une erreur de droits. Le journal des messages du processus indique "Execution was aborted. No security access for ASCIIOutput."

Pour plus de détails sur la manière dont la fonction **Exporter comme données texte** est définie, reportez-vous à la rubrique "Affectation de droits" dans *TM1 Operations*.

**Remarque :** La fonction ASCIIOutput place le caractère hexadécimal 0x1A à la fin de tous les fichiers générés. Cependant, TM1 Web ne peut pas ouvrir une feuille Web contenant le caractère hexadécimal 0x1A.

Si vous utilisez ASCIIOutput pour exporter des données TM1 vers un fichier ASCII, puis tentez d'ouvrir ce fichier dans une feuille Web TM1, cette erreur surviendra.

Error occurred while converting the MS Excel workbook into XML format, hexadécimal value 0x1A is an invalid character.(Erreur pendant la conversion du classeur MS Excel au format XML, la valeur hexadécimale 0x1A est un caractère non valide).

Supprimez le caractère hexadécimal de la feuille Web pour pouvoir ouvrir le fichier dans TM1 Web.

## Syntaxe

```
ASCIIOutput(FileName, String1, String2, ...Stringn);
```

Argument	Description
FileName	Chemin complet du fichier ASCII dans lequel vous voulez écrire l'enregistrement. Ce chemin doit inclure une extension de fichier.
String1...Stringn	Chaîne correspondant à chaque zone à créer dans le fichier ASCII. Cet argument peut être une chaîne ou une variable TurboIntegrator remplaçant une chaîne.

## Exemple

Cet exemple écrit un enregistrement dans le fichier ASCII NewCube.cma. Chaque zone de l'enregistrement correspond à une variable affectée par TurboIntegrator à une colonne de votre source de données.

```
ASCIIOutput('NouveauCube.cma', V1, V2, V3, V4, V5 );
```

## NumberToString

NumberToString convertit un nombre en chaîne en utilisant le séparateur décimal de l'environnement local de l'utilisateur en cours.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Dans Microsoft Windows, le séparateur décimal est une option des paramètres régionaux.

La sortie de cette fonction est similaire au format de nombres 'général'; elle n'utilise pas de séparateur de milliers et utilise le signe moins (-) pour dénoter des nombres négatifs.

## Syntaxe

```
NumberToString(Value);
```

Argument	Description
Value	Valeur réelle à convertir en chaîne.

### Exemple

```
nRET = NumberToString(1234.5);
```

## NumberToStringEx

NumberToStringEx convertit un nombre en chaîne en utilisant le format de chaîne, le séparateur décimal et le séparateur de milliers transmis.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
NumberToStringEx(Value, NumericFormat, DecimalSep, ThousandsSep);
```

Argument	Description
Value	Valeur réelle à convertir en chaîne.
FormatString	Chaîne TM1 au format numérique qui définit le format de la sortie de la fonction. Les formats numériques sont décrits dans la documentation IBM Cognos <i>TM1 Perspectives</i> , <i>TM1 Architect</i> , and <i>TM1 Web</i> .
DecimalSep	Séparateur décimal à utiliser dans la chaîne de sortie.
ThousandsSep	Séparateur de milliers à utiliser dans la chaîne de sortie.

### Exemple

```
sRet=NUMBERTOSTRINGEX(7895.23,'#',0.#####', ','');
```

```
ASCIIOUTPUT('number_to_string.txt',sRet);
```

Will return in ascii file;

7.895,23

## SetInputCharacterSet

Cette fonction permet de spécifier le jeu de caractères à utiliser dans une source de données TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Quand un processus TurboIntegrator lit un fichier externe comme fichier en entrée, il doit connaître le jeu de caractères utilisé lors de l'écriture du fichier externe. Si le fichier contient une marque d'ordre d'octet valide, les fonctions TM1 convertissent correctement le fichier en UTF-8, si nécessaire.

Pour les formats ne contenant pas une marque d'ordre d'octets valide, les caractères doivent être convertis depuis un autre ordre de codage vers UTF-8. Si les convertisseurs appropriés sont présents sur

l'ordinateur hébergeant le serveur, le fichier en entrée est converti pour le jeu de caractères Unicode requis par TM1.

## Syntaxe

```
SetInputCharacterSet (CharacterSet);
```

Argument	Description
CharacterSet	Le codage de caractères dans le fichier en entrée qui sera utilisé par le processus TurboIntegrator. Si l'argument CharacterSet n'est pas un type de caractères connus, le type est réglé par défaut sur celui des paramètres régionaux du système.

Encodage de caractères	Paramètres régionaux du système
TM1CS_ISO_8859_1	ISO-8859-1 Latin-1, Europe occidentale
TM1CS_ISO_8859_2	ISO-8859-2 Latin-2, Europe centrale
TM1CS_ISO_8859_3	ISO-8859-3 Latin-3, Europe méridionale
TM1CS_ISO_8859_4	ISO-8859-4 Latin-4, Europe septentrionale
TM1CS_ISO_8859_5	ISO-8859-5 Latin/Cyrillique
TM1CS_ISO_8859_6	ISO-8859-6 Latin/Arabe
TM1CS_ISO_8859_7	ISO-8859-7 Latin/Grec
TM1CS_ISO_8859_8	ISO-8859-8 Latin/Hébreu
TM1CS_ISO_8859_9	ISO-8859-9 Latin-5, Turc
TM1CS_ISO_8859_10	ISO-8859-10 Latin-6, Nordique
TM1CS_ISO_8859_11	ISO-8859-11 Latin/Thaï
TM1CS_ISO_8859_13	ISO-8859-13 Latin-7, Pays baltes
TM1CS_ISO_8859_14	ISO-8859-14 Latin-8, Celtique
TM1CS_ISO_8859_15	ISO-8859-15 Latin-9, remplace ISO-8859-1
TM1CS_ISO_8859_16	ISO-8859-16 Latin-10, Europe méridionale de l'Est
TM1CS_WCP1250	Microsoft Windows Europe centrale
TM1CS_WCP1251	Windows Cyrillique
TM1CS_WCP1252	Windows Latin-1 multilingue
TM1CS_WCP1253	Windows Grec

Encodage de caractères	Paramètres régionaux du système
TM1CS_WCP1254	Windows Turc
TM1CS_WCP1255	Windows Hébreu
TM1CS_WCP1256	Windows Arabe
TM1CS_WCP1257	Windows Balte
TM1CS_WCP1258	Windows Vietnam
TM1CS_WCP874	Windows Thaï
TM1CS_WCP932	Windows Japonais
TM1CS_WCP936	Windows chinois simplifié
TM1CS_WCP949	Windows coréen
TM1CS_WCP950	Windows chinois traditionnel
TM1CS_KOI8R	Russe et Cyrillique (KOI8-R)
TM1CS_GB18030	UNICODE version PRC
TM1CS_BIG5	Chinois traditionnel
TM1CS_SHIFTJIS	JIS 0201 + JIS 0208, légèrement différent de CP932
TM1CS_SJIS0213	JIS 0213-2004, non BMP requis.
TM1CS_EUC_JP	EUC Japonais
TM1CS_EUC_CN	EUC Chinois simplifié
TM1CS_EUC_KR	EUC Coréen
TM1CS_UTF8	UTF-8
TM1CS_UTF16	UTF-16 Little Endian
TM1CS_UTF16ESC	UNICODE notation
TM1CS_UTF32	UTF-32 Little Endian
TM1CS_OS_DEFAULT	paramètre par défaut du système d'exploitation
TM1CS_LOCALPATH	codage local mais notation UNICODE sur non natif

### Exemple

```
SetInputCharacterSet ('TM1CS_ISO_8859_11');
```

Cet exemple spécifie que le jeu de caractères en entrée pour la source de données TurboIntegrator est ISO-8859-11 Latin/Thai.

## SetOutputCharacterSet

Cette fonction permet de spécifier le jeu de caractères à utiliser pour écrire dans un fichier texte à l'aide de la fonction `TextOutput` dans un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Fonction avec fonction [TextOutput](#).

Utilisée avec des fonctions [TextOutput](#).

### Syntaxe

```
SetOutputCharacterSet( FileName, CharacterSet );
```

Argument	Description
FileName	Chemin complet du fichier texte pour lequel vous souhaitez spécifier un jeu de caractères. Ce chemin doit inclure une extension de fichier.  Cet argument doit être identique à l'argument <code>FileName</code> pour la fonction <a href="#">TextOutput</a> .
CharacterSet	Le codage de caractères à utiliser lors de l'écriture dans le fichier en sortie.

Pour de plus amples informations sur les valeurs acceptables pour `CharacterSet`, voir «[SetInputCharacterSet](#)», à la page 220.

## SetOutputEscapeDoubleQuote

Cette fonction permet de protéger les guillemets qui apparaissent dans des noms d'éléments ou des valeurs de données lors de l'exportation d'une vue de cube dans un fichier `.csv`.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Lorsque `SetOutputEscapeDoubleQuote` est inclus dans le script TurboIntegrator avec la valeur 1, le fichier exporté conserve la position des guillemets doubles tels qu'ils apparaissent dans la vue de cube source en insérant une paire supplémentaire de guillemets doubles autour de chaque guillemet double comme caractère d'échappement. Par exemple, si le nom d'un élément de la vue source est "Région", il est exporté dans le fichier de sortie `.csv` sous la forme ""Région"".

Lorsque `SetOutputEscapeDoubleQuote` n'est *pas* inclus dans le script TurboIntegrator, ou si sa valeur est 0, le fichier exporté n'échappe aucun guillemet double du cube source.

`SetOutputEscapeDoubleQuote` est utilisé avec la fonction `ASCIIOutput`, qui écrit effectivement le fichier en sortie. `SetOutputEscapeDoubleQuote` doit précéder `ASCIIOutput` dans le script TurboIntegrator, et les deux fonctions doivent utiliser la même valeur pour le paramètre `FileName`.

### Syntaxe

```
SetOutputEscapeDoubleQuote(FileName, Num);
```

Argument	Description
FileName	Chemin complet du fichier dans lequel vous voulez écrire la vue de cube. Ce chemin doit inclure une extension de fichier.
Num	Indicateur de l'échappement des guillemets doubles dans le fichier en sortie.  1 indique que les guillemets doubles seront échappés dans le fichier en sortie.  0 indique que les guillemets doubles ne seront pas échappés dans le fichier en sortie.

### Exemple

```
SetOutputEscapeDoubleQuote('C:\temp\cube1.csv', 1);
```

Dans cet exemple, tous les guillemets doubles rencontrés dans la vue du cube source lors de l'écriture de la sortie dans le fichier C:\temp\cube1.csv sont échappés.

## StringToNumber

StringToNumber convertit une chaîne en nombre en utilisant le séparateur décimal de l'environnement local de l'utilisateur en cours. Si la chaîne en entrée est une chaîne numérique non valide, la valeur renvoyée sera une valeur à virgule flottante non valide. Dans Microsoft Windows, le séparateur décimal est une option des paramètres régionaux.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
StringToNumber(String);
```

Argument	Description
String	Chaîne à convertir en nombre.

### Exemple

```
nRET = StringToNumber('123.45');
```

## StringToNumberEx

StringToNumberEx convertit une chaîne en nombre en utilisant le format de chaîne, le séparateur décimal et le séparateur de milliers transmis. Si la chaîne en entrée est une chaîne numérique non valide, la valeur renvoyée sera une valeur à virgule flottante non valide.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
StringToNumberEx(String, DecimalSep, ThousandsSep);
```



Argument	Description
String	Chaîne à convertir en nombre.
DecimalSep	Séparateur décimal à utiliser dans le nombre en sortie.
ThousandsSep	Séparateur de milliers à utiliser dans le nombre en sortie.

### Exemple

```
nRET = StringToNumberEx('12453.45', ' . ', ' ', '');
```

## TextOutput

Cette fonction écrit un enregistrement séparé par des virgules dans un fichier texte.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, TextOutput utilise le jeu de caractères des paramètres régionaux de l'ordinateur du serveur. Pour créer un fichier utilisant un autre jeu de caractères, appelez la fonction [SetOutputCharacterSet](#) avant d'appeler TextOutput.

Le fichier texte est ouvert lors de l'écriture du premier enregistrement et fermé à la fin de la procédure TurboIntegrator (Prologue, Métadonnées, Données, Epilogue) contenant la fonction TextOutput.

Si vous utilisez la fonction TextOutput pour écrire dans le même fichier dans plusieurs procédures (onglets) d'un processus TurboIntegrator, ce fichier est remplacé chaque fois qu'il est ouvert pour une nouvelle procédure.

Tous les enregistrements en sortie générés par TextOutput sont limités à 8000 octets. Si un enregistrement en sortie dépasse 8000 octets, cet enregistrement est tronqué et un avertissement est écrit dans le fichier TM1ProcessError.log.

Quand TextOutput rencontre un argument chaîne qui pousse l'enregistrement en sortie au-delà de la limite de 8000 octets, il ignore cet argument et tous les arguments qui suivent. Par exemple, s'il y a 10 arguments chaînes et que la sortie pour les sept premiers arguments atteint 7950 octets alors que la sortie pour le huitième argument est de 51 octets, seule la sortie des sept premiers arguments est écrite dans l'enregistrement. S'il y a dix arguments chaînes et que le premier argument dépasse 8000 octets, aucune sortie n'est écrite dans l'enregistrement.

La fonction TextOutput génère une erreur mineure si une erreur se produit lors de l'écriture du fichier ASCII. En outre, la fonction renvoie une valeur de l'exécution : 1 si la fonction écrit avec succès le fichier texte et 0 en cas d'échec.

L'erreur sera générée et la valeur renvoyée uniquement quand TextOutput écrit sur un disque autre que celui où s'exécute le serveur. Par exemple, si le serveur tourne sur le lecteur C:, que TextOutput écrit sur le lecteur F: et que F: vient à manquer d'espace, l'erreur est piégée et le serveur reste actif. Par exemple, si le serveur s'exécute sur le lecteur C: alors que TextOutput écrit également sur cette unité et que cette unité vient à manquer d'espace, le serveur est arrêté (comme prévu).

**Remarque :** La possibilité d'exécuter ou non la fonction TextOutput lorsque la source de données est une vue de cube est déterminée par l'affectation de la fonction **Exporter comme données texte**, qui est définie par groupe d'utilisateurs. Si un utilisateur est membre d'un groupe qui n'est pas autorisé à exporter les données sous forme de texte, toute tentative par cet utilisateur d'exécuter TextOutput entraîne l'arrêt du processus avec une erreur de droits. Le journal des messages du processus indique "Execution was aborted. No security access for TextOutput."

Pour plus de détails sur la manière dont la fonction **Exporter comme données texte** est définie, reportez-vous à la rubrique "Affectation de droits" du document *IBM Cognos TM1 Operations*.

## Syntaxe

```
TextOutput(FileName, String1, String2, ...Stringn);
```

Argument	Description
FileName	Chemin complet du fichier texte dans lequel vous voulez écrire l'enregistrement. Ce chemin doit inclure une extension de fichier.
String1...Stringn	Chaîne correspondant à chaque zone à créer dans le fichier texte. Cet argument peut être une chaîne ou une variable TurboIntegrator remplaçant une chaîne.

## Exemple

```
TextOutput('NouveauCube.cma', V1, V2, V3, V4, V5 );
```

Cet exemple écrit un enregistrement dans le fichier NouveauCube.cma. Chaque zone de l'enregistrement correspond à une variable affectée par TurboIntegrator à une colonne de votre source de données.

## Fonctions TurboIntegrator de manipulation d'attributs

Ces fonctions facilitent la manipulation des attributs.

### ATTRNL

ATTRNL renvoie un attribut numérique pour un élément spécifié d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ATTRNL(DimName, ElName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
ElName	Élément de la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code>, si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Engine Size de l'élément L Series 1.8L Sedan dans la dimension Model pour l'environnement local français.

```
ATTRNL('Model', 'L Series 1.8L Sedan', 'Engine Size', 'fr');
```

## ATTRSL

AttrSL renvoie un attribut de chaîne pour un élément spécifié d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AttrSL(DimName, ElName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
ElName	Élément de la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est fr-CA, la fonction renvoie la valeur de l'attribut pour l'environnement local fr-CA, si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour fr-CA, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local fr parent. Si la valeur de l'attribut n'est pas disponible pour fr, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Currency de l'élément 10100 dans la dimension Plan\_Business\_Unit pour l'environnement local français.

```
AttrSL('Plan_Business_Unit', '10100', 'Currency', 'fr');
```

### AttrDelete

AttrDelete supprime un attribut d'élément de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AttrDelete(DimName, AttrName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez supprimer un attribut d'élément.
AttrName	Nom de l'attribut à supprimer.

## Exemple

Cet exemple supprime l'attribut d'élément Couleurintérieure pour la dimension Modèle.

```
AttrDelete('Modèle', 'Couleurintérieure');
```

## AttrInsert

Cette fonction crée un attribut d'élément pour une dimension. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AttrInsert(DimName, PrevAttr, AttrName, Type);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un attribut d'élément.
PrevAttr	Attribut précédant l'attribut que vous créez.
AttrName	Nom à affecter au nouvel attribut.
Type	Type d'attribut. Il existe trois valeurs possibles pour l'argument Type : <ul style="list-style-type: none"><li>• N - Crée un attribut numérique.</li><li>• S - Crée un attribut chaîne.</li><li>• A - Crée un attribut alias.</li></ul>

## Exemple

Cet exemple crée l'attribut de chaîne InteriorColor pour la dimension Model. Cet attribut est inséré à la suite de l'attribut Transmission.

```
AttrInsert('Model', 'Transmission', 'InteriorColor', 'S');
```

## AttrPutN

Cette fonction affecte une valeur à un attribut d'élément numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AttrPutN( Value, DimName, EName, AttrName, [LangLocaleCode] );
```

Argument	Description
Value	Valeur numérique à affecter à un attribut d'élément.
DimName	Dimension parent de l'élément pour lequel vous voulez affecter une valeur d'attribut.

Argument	Description
ElName	Élément pour lequel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 2257993 à l'attribut Codeproduit de l'élément Berline 1,8L Série S dans la dimension Modèle.

```
AttrPutN(2257993, 'Modèle', 'Berline 1,8L Série S', 'Codeproduit');
```

### AttrPutS

Cette fonction affecte une valeur à un attribut d'élément de chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AttrPutS(Value, DimName, ElName, AttrName, [LangLocaleCode] );
```

Argument	Description
Value	Valeur à affecter à un attribut d'élément.
DimName	Dimension parent de l'élément pour lequel vous voulez affecter une valeur d'attribut.
ElName	Élément pour lequel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la chaîne de valeur Beige à l'attribut InteriorColor de l'élément S Series 1.8L Sedan dans la dimension Model.

```
AttrPutS('Beige','Modèle', 'Berline 1,8L Série S', 'Couleurintérieure');
```

### ChoreAttrDelete

ChoreAttrDelete supprime un attribut de tâche de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ChoreAttrDelete(AttrName);
```

Argument	Description
AttrName	Nom de l'attribut de tâche à supprimer.

### Exemple

Cet exemple supprime l'attribut Description des tâches de votre serveur TM1.

```
ChoreAttrDelete('Description');
```

### ChoreAttrInsert

ChoreAttrInsert crée un nouvel attribut pour les tâches de votre serveur TM1. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de tâche existant, commencez par le supprimer à l'aide de la fonction ChoreAttrDelete. Vous pouvez ensuite utiliser ChoreAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

## Syntaxe

```
ChoreAttrInsert( PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent, ou si vous souhaitez que le nouvel attribut soit le premier attribut des tâches, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de tâche.
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"><li>• N - Crée un attribut numérique.</li><li>• S - Crée un attribut chaîne.</li><li>• A - Crée un attribut alias.</li></ul>

## Exemple

Cet exemple crée l'attribut de type chaîne Description pour les tâches. Cet attribut est inséré à la suite de l'attribut Owner.

```
ChoreAttrInsert('Owner', 'Description', 'S');
```

## ChoreAttrN

ChoreAttrN renvoie un attribut numérique d'une tâche donnée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ChoreAttrN(ChoreName, AttrName);
```

Argument	Description
ChoreName	Nom de tâche valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la tâche.

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Division\_Code de la tâche Import.

```
ChoreAttrN('Import', 'Division_Code');
```

## ChoreAttrNL

ChoreAttrNL renvoie la valeur numérique d'un attribut pour une tâche spécifiée, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



## Syntaxe

```
ChoreAttrNL(ChoreName, AttrName, [LangLocaleCode]);
```

Argument	Description
ChoreName	Nom de tâche valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la tâche.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Division_Code` de la tâche `Import` pour l'environnement local français.

```
ChoreAttrNL('Import', 'Division_Code', 'fr');
```

## ChoreAttrPutN

Cette fonction affecte une valeur à un attribut de tâche numérique.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

## Syntaxe

```
ChoreAttrPutN(NumericValue, ChoreName, AttrName, [LangLocaleCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de tâche.
ChoreName	Tâche à laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 7161994 à l'attribut Division\_Code de la tâche Import pour le code d'environnement local de langue français.

```
ChoreAttrPutN(7161994, 'Import', 'Division_Code','fr');
```

### ChoreAttrPutS

Cette fonction affecte une valeur à un attribut de tâche chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ChoreAttrPutS(String, ChoreName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de tâche.
ChoreName	Tâche à laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur de type chaîne Ricci à l'attribut Owner de la tâche Import pour le code d'environnement local de langue français.

```
ChoreAttrPutS('Ricci', 'Import', 'Owner', 'fr');
```

## ChoreAttrS

ChoreAttrS renvoie un attribut de type chaîne pour une tâche donnée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ChoreAttrS(ChoreName, AttrName);
```

Argument	Description
ChoreName	Nom de tâche valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la tâche.

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner de la tâche Exchange\_Rate\_Updates.

```
ChoreAttrS('Exchange_Rate_Updates', 'Owner');
```

## ChoreAttrSL

ChoreAttrSL renvoie la valeur de chaîne d'un attribut pour une tâche spécifique, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ChoreAttrSL(ChoreName, AttrName, [LangLocaleCode]);
```

Argument	Description
ChoreName	Nom de tâche valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la tâche.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner de la tâche Depreciate\_Inventory pour l'environnement local français.

```
ChoreAttrSL('Depreciate_Inventory', 'Owner', 'fr');
```

### CreateHierarchyByAttribute

CreateHierarchyByAttribute crée une hiérarchie simple à 3 niveaux depuis un attribut unique.

La nouvelle hiérarchie se compose d'un élément racine de niveau supérieur, d'un niveau intermédiaire de consolidations représentant les valeurs d'attribut existantes, et d'un niveau inférieur de feuilles de dimension incluant la valeur d'attribut associée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

**Remarque :** cette fonction crée une hiérarchie depuis l'ensemble en cours de valeurs d'attribut, mais le système ne synchronise pas automatiquement la hiérarchie avec les données d'attribut lorsque celles-ci changent. Les modélisateurs doivent régénérer la hiérarchie en fonction des besoins.

## Syntaxe

```
CreateHierarchyByAttribute(DimName, AttrName [, emptyParent [, rootName ] ] );
```

Argument	Description
DimName	Nom de la dimension contenant l'attribut. Une hiérarchie du même nom que la dimension est créée.
AttrName	Nom de l'attribut depuis lequel créer la hiérarchie.
emptyParent	Spécifie le nom d'une consolidation à créer, qui collecte des feuilles de dimension ne possédant pas de valeur d'attribut. S'il s'agit d'une chaîne vide, la fonction ne crée pas de consolidation.
rootName	Remplace le nom de l'élément racine qui est dérivé par défaut de l'attribut.

## Exemple

```
CreateHierarchyByAttribute ('Country', 'City');
```

Cet exemple crée une hiérarchie depuis l'attribut City dans la dimension Country.

## CubeAttrDelete

CubeAttrDelete supprime un attribut de cube de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeAttrDelete(AttrName);
```

Argument	Description
AttrName	Nom de l'attribut de cube à supprimer.

## Exemple

Cet exemple supprime l'attribut Description des cubes de votre serveur TM1.

```
CubeAttrDelete('Description');
```

## CubeAttrInsert

CubeAttrInsert crée un nouvel attribut pour les cubes de votre serveur TM1. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de cube existant, commencez par le supprimer à l'aide de la fonction CubeAttrDelete. Vous pouvez ensuite utiliser CubeAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

### Syntaxe

```
CubeAttrInsert( PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent, ou si vous souhaitez que le nouvel attribut soit le premier attribut des cubes, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de cube.
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"> <li>• N - Crée un attribut numérique.</li> <li>• S - Crée un attribut chaîne.</li> <li>• A - Crée un attribut alias.</li> </ul>

### Exemple

Cet exemple crée l'attribut de type chaîne Description pour les cubes. Cet attribut est inséré à la suite de l'attribut Owner.

```
CubeAttrInsert('Owner', 'Description', 'S');
```

### CubeAttrPutN

CubeAttrPutN affecte une valeur à un attribut de cube numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeAttrPutN(NumericValue, CubeName, AttrName, [LangLocaleCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de cube.
CubeName	Cube auquel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 07161994 à l'attribut AccountingCode du cube Sales pour le code d'environnement local de langue français.

```
CubeAttrPutN(07161994, 'Sales', 'AccountingCode', 'fr');
```

### CubeAttrPutS

CubeAttrPutS affecte une valeur à un attribut de cube de type chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeAttrPutS(String, CubeName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de cube.
CubeName	Cube auquel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

## Exemple

Cet exemple affecte la valeur de type chaîne Prototype à l'attribut Description du cube Sales pour le code d'environnement local de langue français.

```
CubeAttrPutS('Prototype', 'Sales', 'Description','fr');
```

## CubeATTRNL

CubeATTRNL renvoie la valeur numérique d'un attribut pour un cube particulier, en fonction d'un environnement local donné.

Cette fonction est valide dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeATTRNL(CubeName, AttrName, [LangLocaleCode]);
```

Argument	Description
CubeName	Nom de cube valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide du cube.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est fr-CA, la fonction renvoie la valeur de l'attribut pour l'environnement local fr-CA (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour fr-CA, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local fr (français) parent. Si la valeur de l'attribut n'est pas disponible pour fr, l'attribut de base est renvoyé.</p>



## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code du cube Product pour l'environnement local français.

```
CubeATTRNL('Product', 'Accounting_Code', 'fr');
```

## CubeATTRSL

CubeATTRSL renvoie la valeur de type chaîne d'un attribut pour un cube particulier, en fonction d'un environnement local donné.

Cette fonction est valide dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeATTRSL(CubeName, AttrName, [LangLocaleCode]);
```

Argument	Description
CubeName	Nom de cube valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide du cube.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est fr-CA, la fonction renvoie la valeur de l'attribut pour l'environnement local fr-CA (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour fr-CA, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local fr (français) parent. Si la valeur de l'attribut n'est pas disponible pour fr, l'attribut de base est renvoyé.</p>

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner du cube Product pour l'environnement local français.

```
CubeATTRSL('Product', 'Owner', 'fr');
```

## DimensionAttrDelete

DimensionAttrDelete supprime un attribut de dimension de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionAttrDelete(AttrName);
```

Argument	Description
AttrName	Nom de l'attribut de dimension à supprimer.

## Exemple

Cet exemple supprime l'attribut Description des dimensions de votre serveur TM1.

```
DimensionAttrDelete('Description');
```

## DimensionAttrInsert

DimensionAttrInsert crée un nouvel attribut pour les dimensions de votre serveur TM1. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de dimension existant, commencez par le supprimer à l'aide de la fonction DimensionAttrDelete. Vous pouvez ensuite utiliser DimensionAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

### Syntaxe

```
DimensionAttrInsert( PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent, ou si vous souhaitez que le nouvel attribut soit le premier attribut des dimensions, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de dimension.

Argument	Description
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"> <li>• N - Crée un attribut numérique.</li> <li>• S - Crée un attribut chaîne.</li> <li>• A - Crée un attribut alias.</li> </ul>

### Exemple

Cet exemple crée l'attribut de type chaîne Description pour les dimensions. Le paramètre PrevAttrName étant absent, cet attribut est inséré comme premier attribut des dimensions de votre serveur TM1.

```
DimensionAttrInsert('', 'Description', 'S');
```

### DimensionAttrPutN

DimensionAttrPutN affecte une valeur à un attribut de dimension numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionAttrPutN(NumericValue, DimensionName, AttrName, [LocalLangCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.  Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.  Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.

### Exemple

Cet exemple affecte la valeur 07161994 à l'attribut AccountingCode de la dimension Models pour le code d'environnement local de langue français.

```
DimensionAttrPutN(07161994, 'Models', 'AccountingCode', 'fr');
```

## DimensionAttrPutS

DimensionAttrPutS affecte une valeur à un attribut de dimension de type chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionAttrPutS(String, DimensionName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur de type chaîne Prototype à l'attribut Description de la dimension Model pour le code d'environnement local de langue français.

```
DimensionAttrPutS('Prototype', 'Model', 'Description', 'fr');
```

## DimensionATTRNL

DimensionATTRNL renvoie la valeur numérique d'un attribut pour une dimension particulière, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionATTRNL(DimName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Accounting_Code` de la dimension `Plan_Business_Unit` pour l'environnement local français.

```
DimensionATTRNL('Plan_Business_Unit', 'Accounting_Code', 'fr');
```

### DimensionATTRSL

`DimensionATTRSL` renvoie la valeur de type chaîne d'un attribut pour une dimension particulière, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionATTRSL(DimName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Manager de la dimension Plan\_Business\_Unit pour l'environnement local français.

```
DimensionATTRSL('Plan_Business_Unit', 'Manager', 'fr');
```

### ElementATTRNL

ElementATTRNL renvoie un attribut numérique pour un élément spécifié d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementATTRNL(DimName, HierName, ElName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément de la dimension.

Argument	Description
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code>, si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Engine Size de l'élément L Series 1.8L Sedan dans la dimension Model pour l'environnement local français. Cet exemple s'applique à la hiérarchie 2015.

```
ATTRNL('Model', '2015', 'L Series 1.8L Sedan', 'Engine Size', 'fr');
```

### ElementATTRSL

ElementATTRSL renvoie un attribut de chaîne pour un élément spécifié d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementATTRSL(DimName, HierName, ElName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.

Argument	Description
ElName	Elément de la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de l'élément.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code>, si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Currency de l'élément 10100 dans la dimension Plan\_Business\_Unit pour l'environnement local français.

```
ElementATTRSL('Plan_Business_Unit', '10100', 'Currency', 'fr');
```

### ElementAttrPutN

Cette fonction affecte une valeur à un attribut d'élément numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementAttrPutN( Value, DimName, HierName, ElName, AttrName, [LangLocaleCode] );
```

Argument	Description
Value	Valeur numérique à affecter à un attribut d'élément.



Argument	Description
DimName	Dimension parent de l'élément pour lequel vous voulez affecter une valeur d'attribut.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément pour lequel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 2257993 à l'attribut ProdCode de l'élément S Series 1.8L Sedan dans la hiérarchie Automobile de la dimension Model.

```
ElementAttrPutN(2257993, 'Model', 'Automobile', ' S Series 1.8L Sedan ', 'ProdCode');
```

### ElementAttrPutS

Cette fonction affecte une valeur à un attribut d'élément chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementAttrPutS(Value, DimName, HierName, ElName, AttrName, [LangLocaleCode] );
```

Argument	Description
Value	Valeur à affecter à un attribut d'élément.
DimName	Dimension parent de l'élément pour lequel vous voulez affecter une valeur d'attribut.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément pour lequel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la chaîne de valeur Beige à l'attribut InteriorColor de l'élément S Series 1.8L Sedan dans la hiérarchie Automobile de la dimension Model.

```
ElementAttrPutS('Beige', 'Model', 'Automobile', 'S Series 1.8L Sedan', 'InteriorColor');
```

## ElementAttrInsert

ElementAttrInsert crée un attribut d'élément pour une dimension. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementAttrInsert(DimName, HierName, PrevAttr, AttrName, Type);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un attribut d'élément.
HierName	Nom de la hiérarchie dans la dimension.
PrevAttr	Attribut précédant l'attribut que vous créez.
AttrName	Nom à affecter au nouvel attribut.
Type	<p>Type d'attribut. Il existe trois valeurs possibles pour l'argument Type :</p> <ul style="list-style-type: none"> <li>• N - Crée un attribut numérique.</li> <li>• S - Crée un attribut chaîne.</li> <li>• A - Crée un attribut alias.</li> </ul>

### Exemple

Cet exemple crée l'attribut de chaîne InteriorColor dans la hiérarchie Automobile de la dimension Model. Cet attribut est inséré à la suite de l'attribut Transmission.

```
ElementAttrInsert('Model', 'Automobile', 'Transmission', 'InteriorColor', 'S');
```

## ElementAttrDelete

ElementAttrDelete supprime un attribut d'élément de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ElementAttrDelete(DimName, HierName, AttrName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez supprimer un attribut d'élément.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Nom de l'attribut à supprimer.

### Exemple

Cet exemple supprime l'attribut d'élément InteriorColor de la hiérarchie Automobile dans la dimension Model.

```
ElementAttrDelete('Model', 'Automobile', 'InteriorColor');
```

## HierarchyAttrPutN

HierarchyAttrPutN affecte une valeur à un attribut numérique dans une hiérarchie spécifiée d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyAttrPutN(NumericValue, DimensionName, HierName, AttrName, [LocalLangCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 07161994 à l'attribut AccountingCode de la dimension Models pour le code d'environnement local de langue français. Cette modification est appliquée à la dimension Receivables dans la dimension Models.

```
HierarchyAttrPutN(07161994, 'Models', 'Receivables', 'AccountingCode', 'fr');
```

## HierarchyAttrPutS

HierarchyAttrPutS affecte une valeur à un attribut de chaîne dans une hiérarchie spécifiée d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyAttrPutS(String, DimensionName, HierName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

## Exemple

Cet exemple affecte la valeur de type chaîne Prototype à l'attribut Description de la dimension Model pour le code d'environnement local de langue français. Cette modification est appliquée à la dimension Receivables dans la dimension Model.

```
HierarchyAttrPutS('Prototype', 'Model', 'Receivables', 'Description','fr');
```

## HierarchyATTRN

HierarchyATTRN renvoie un attribut numérique pour une hiérarchie spécifiée d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyATTRN(DimName, HierName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code de la dimension Plan\_Business\_Unit. Cet exemple s'applique à la hiérarchie Equipment.

```
HierarchyATTRN('Plan_Business_Unit', 'Equipment', 'Accounting_Code');
```

## HierarchyATTRS

HierarchyATTRS renvoie un attribut de chaîne pour une hiérarchie spécifiée d'une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyATTRS(DimName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Manager de la dimension Plan\_Business\_Unit. Cet exemple s'applique à la hiérarchie Equipment.

```
HierarchyATTRS('Plan_Business_Unit', 'Equipment', 'Manager');
```

## HierarchyATTRNL

HierarchyATTRNL renvoie la valeur numérique d'un attribut pour une hiérarchie spécifiée d'une dimension, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyATTRNL(DimName, HierName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est fr-CA, la fonction renvoie la valeur de l'attribut pour l'environnement local fr-CA (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour fr-CA, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local fr (français) parent. Si la valeur de l'attribut n'est pas disponible pour fr, l'attribut de base est renvoyé.</p>

## Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code de la dimension Plan\_Business\_Unit pour l'environnement local français. Cette fonction s'applique à la hiérarchie Equipment.

```
HierarchyATTRNL('Plan_Business_Unit', 'Equipment', 'Accounting_Code', 'fr');
```

## HierarchyATTRSL

HierarchyATTRSL renvoie la valeur de type chaîne d'un attribut pour une hiérarchie spécifiée d'une dimension, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyATTRSL(DimName, HierName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom de la hiérarchie dans la dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est fr-CA, la fonction renvoie la valeur de l'attribut pour l'environnement local fr-CA (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour fr-CA, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local fr (français) parent. Si la valeur de l'attribut n'est pas disponible pour fr, l'attribut de base est renvoyé.</p>

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Manager de la dimension Plan\_Business\_Unit pour l'environnement local français. Cette fonction s'applique à la hiérarchie Equipment.

```
HierarchyATTRSL('Plan_Business_Unit', 'Equipment', 'Manager', 'fr');
```

## HierarchySubsetATTRS

HierarchySubsetATTRS renvoie un attribut de chaîne pour un sous-ensemble spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetATTRS(DimName, HierName, SubName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

## Exemple

Dans cet exemple, la fonction renvoie la valeur de chaîne de l'attribut Manager du sous-ensemble Sales de la hiérarchie Europe dans la dimension Plan\_Business\_Unit.

```
HierarchySubsetATTRS('Plan_Business_Unit', 'Europe', 'Sales', 'Manager');
```

## HierarchySubsetATTRN

HierarchySubsetATTRN renvoie un attribut numérique pour un sous-ensemble spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetATTRN(DimName, HierName, SubName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.



Argument	Description
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code du sous-ensemble Sales de la hiérarchie Europe dans la dimension Plan\_Business\_Unit.

```
HierarchySubsetATTRN('Plan_Business_Unit', 'Europe', 'Sales', 'Accounting_Code');
```

### HierarchySubsetATTRSL

HierarchySubsetATTRSL renvoie la valeur de chaîne d'un attribut pour un sous-ensemble (et un environnement local) spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetATTRSL(DimName, HierName, SubName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de chaîne de l'attribut Manager du sous-ensemble Sales (de la hiérarchie Europe) pour l'environnement local français.

```
HierarchySubsetATTRSL('Plan_Business_Unit', 'Europe', 'Sales', 'Manager', 'fr');
```

## HierarchySubsetATTRNL

HierarchySubsetATTRNL renvoie la valeur numérique d'un attribut pour un sous-ensemble (et un environnement local) spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetATTRNL(DimName, HierName, SubName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.

Argument	Description
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Accounting_Code` du sous-ensemble `Sales` (de la hiérarchie `Europe`) pour l'environnement local français.

```
HierarchySubsetATTRNL('Plan_Business_Unit', 'Europe', 'Sales', 'Accounting_Code', 'fr');
```

### HierarchySubsetAttrPutS

`HierarchySubsetAttrPutS` affecte une valeur de chaîne à un attribut d'un sous-ensemble spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

### Syntaxe

```
HierarchySubsetAttrPutS(String, DimName, HierName, SubName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de dimension.

Argument	Description
DimName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur de type chaîne Prototype à l'attribut Description du sous-ensemble Z (de la hiérarchie 2016 dans la dimension Model) pour le code d'environnement local de langue français.

```
HierarchySubsetAttrPutS('Prototype', 'Model', '2016', 'Z', 'Description','fr');
```

### HierarchySubsetAttrPutN

HierarchySubsetAttrPutN affecte une valeur numérique à un attribut d'un sous-ensemble spécifié associé à une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetAttrPutN(NumericValue, DimName, HierName, SubName, AttrName, [LocalLangCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de dimension.
DimName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
HierName	Nom d'une hiérarchie dans une dimension.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 07161994 à l'attribut AccountingCode du sous-ensemble Z (de la hiérarchie 2016 dans la dimension Models) pour le code d'environnement local de langue français.

```
HierarchySubsetAttrPutN(07161994, 'Models', '2016', 'Z', 'AccountingCode','fr');
```

## HierarchySubsetAttrInsert

HierarchySubsetAttrInsert crée un nouvel attribut pour les sous-ensembles de votre serveur TM1. Cette fonction crée un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe



**Avertissement :** Pour mettre à jour un attribut de sous-ensemble existant, commencez par le supprimer à l'aide de la fonction HierarchySubsetAttrDelete. Vous pouvez ensuite utiliser HierarchySubsetAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

```
HierarchySubsetAttrInsert( Dimension, Hierarchy, PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
Dimension	Nom de la dimension dont les sous-ensembles sont en cours de mise à jour.
Hierarchy	Nom d'une hiérarchie dans une dimension.
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent ou que vous souhaitez que le nouvel attribut soit le premier attribut des sous-ensembles, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de sous-ensemble.

Argument	Description
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"> <li>• N - Crée un attribut numérique.</li> <li>• S - Crée un attribut chaîne.</li> <li>• A - Crée un attribut alias.</li> </ul>

### Exemple

Cet exemple crée l'attribut de chaîne Description pour les sous-ensembles de la hiérarchie Z de la dimension Model. Le paramètre PrevAttrName étant absent, cet attribut est inséré comme premier attribut des sous-ensembles sur votre serveur TM1.

```
HierarchySubsetAttrInsert('Model', 'Z', '', 'Description', 'S');
```

### HierarchySubsetAttrDelete

HierarchySubsetAttrDelete supprime un attribut de sous-ensemble de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetAttrDelete(Dimension, Hierarchy, AttrName);
```

Argument	Description
Dimension	Nom de la dimension dont l'attribut de sous-ensemble est en cours de suppression.
Hierarchy	Nom d'une hiérarchie dans une dimension.
AttrName	Nom de l'attribut de dimension à supprimer.

### Exemple

Cet exemple supprime l'attribut Description des sous-ensembles de la hiérarchie Z dans la dimension Model.

```
HierarchySubsetAttrDelete('Model', 'Z', 'Description');
```

### ProcessAttrDelete

ProcessAttrDelete supprime un attribut de processus de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrDelete(AttrName);
```

Argument	Description
AttrName	Nom de l'attribut de processus à supprimer.

## Exemple

Cet exemple supprime l'attribut Description des processus de votre serveur TM1.

```
ProcessAttrDelete('Description');
```

## ProcessAttrInsert

ProcessAttrInsert crée un nouvel attribut pour les processus de votre serveur TM1. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de processus existant, commencez par le supprimer à l'aide de la fonction ProcessAttrDelete. Vous pouvez ensuite utiliser ProcessAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

## Syntaxe

```
ProcessAttrInsert( PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent, ou si vous souhaitez que le nouvel attribut soit le premier attribut des processus, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de processus.
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"><li>• N - Crée un attribut numérique.</li><li>• S - Crée un attribut chaîne.</li><li>• A - Crée un attribut alias.</li></ul>

## Exemple

Cet exemple crée l'attribut de type chaîne Description pour les processus. Cet attribut est inséré à la suite de l'attribut Owner.

```
ProcessAttrInsert('Owner', 'Description', 'S');
```

## ProcessAttrN

ProcessAttrN renvoie un attribut numérique d'un processus donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ProcessAttrN(ProcessName, AttrName);
```

Argument	Description
ProcessName	Nom de processus valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide du processus.

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Store\_Code du processus Daily\_Sales.

```
ProcessAttrN('Daily_Sales', 'Store_Code');
```

### ProcessAttrNL

ProcessAttrNL renvoie la valeur numérique d'un attribut pour un processus spécifié, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrNL(ProcessName, AttrName, [LangLocaleCode]);
```

Argument	Description
ProcessName	Nom de processus valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide du processus.



Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Store_Code` du processus `Daily_Sales` pour l'environnement local français.

```
ProcessAttrNL('Daily_Sales', 'Store_Code', 'fr');
```

### ProcessAttrPutN

Cette fonction affecte une valeur à un attribut de processus numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrPutN(NumericValue, CubeName, AttrName, [LangLocaleCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de processus.
ProcessName	Processus auquel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 8051997 à l'attribut Store\_Code du processus Daily\_Sales pour le code d'environnement local de langue français.

```
ProcessAttrPutN(8051997, 'Daily_Sales', 'Store_Code', 'fr');
```

### ProcessAttrPutS

Cette fonction affecte une valeur à un attribut de processus chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrPutS(String, ProcessName, AttrName, [LangLocaleCode] );
```

Argument	Description
Chaîne	Chaîne à affecter à un attribut de processus.
ProcessName	Processus auquel vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

## Exemple

Cet exemple affecte la valeur de type chaîne Ricci à l'attribut Owner du processus Import\_Transactional pour le code d'environnement local de langue français.

```
ProcessAttrPutS('Ricci', 'Import_transactional', 'Owner', 'fr');
```

## ProcessAttrS

ProcessAttrS renvoie un attribut de type chaîne pour un processus donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrS(ProcessName, AttrName);
```

Argument	Description
ProcessName	Nom de processus valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide du processus.

## Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner du processus Refresh\_Cubes.

```
ProcessAttrS('Refresh_Cubes', 'Owner');
```

## ProcessAttrSL

ProcessAttrSL renvoie la valeur de type chaîne d'un attribut pour un processus particulier, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessAttrSL(ProcessName, AttrName, [LangLocaleCode]);
```

Argument	Description
ProcessName	Nom de processus valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide du processus.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Owner du processus Exchange\_Rate\_Update pour l'environnement local français (Canada).

```
ProcessAttrSL('Exchange_Rate_Update', 'Owner', 'fr-CA');
```

### SubsetATTRS

SubsetATTRS renvoie un attribut de type chaîne pour un sous-ensemble donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetATTRS(DimName, SubName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
SubName	Nom d'un sous-ensemble dans une dimension.

Argument	Description
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

### Exemple

Dans cet exemple, la fonction renvoie la valeur de chaîne de l'attribut Manager du sous-ensemble Sales de la dimension Plan\_Business\_Unit.

```
SubsetATTRS('Plan_Business_Unit', 'Sales', 'Manager');
```

## SubsetATTRN

SubsetATTRN renvoie un attribut numérique d'un sous-ensemble donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetATTRN(DimName, SubName, AttrName);
```

Argument	Description
DimName	Nom de dimension valide.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Accounting\_Code du sous-ensemble Sales de la dimension Plan\_Business\_Unit.

```
SubsetATTRN('Plan_Business_Unit', 'Sales', 'Accounting_Code');
```

## SubsetATTRSL

SubsetATTRSL renvoie la valeur de chaîne d'un attribut pour un sous-ensemble spécifié, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetATTRSL(DimName, SubName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.
SubName	Nom d'un sous-ensemble dans une dimension.

Argument	Description
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de chaîne de l'attribut Manager du sous-ensemble Sales pour l'environnement local français.

```
SubsetATTRSL('Plan_Business_Unit', 'Sales', 'Manager', 'fr');
```

### SubsetATTRNL

SubsetATTRNL renvoie la valeur numérique d'un attribut pour un sous-ensemble spécifié, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetATTRNL(DimName, SubName, AttrName, [LangLocaleCode]);
```

Argument	Description
DimName	Nom de dimension valide.

Argument	Description
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez obtenir la valeur. L'argument doit être un attribut valide de la dimension.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Accounting_Code` du sous-ensemble `Sales` pour l'environnement local français.

```
SubsetATTRNL('Plan_Business_Unit', 'Sales', 'Accounting_Code', 'fr');
```

### SubsetAttrPutS

SubsetAttrPutS affecte une valeur de chaîne à un attribut d'un sous-ensemble spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetAttrPutS(String, DimensionName, SubName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur de type chaîne Prototype à l'attribut Description du sous-ensemble Z (de la dimension Model) pour le code d'environnement local de langue français.

```
SubsetAttrPutS('Prototype', 'Model', 'Z', 'Description','fr');
```

### SubsetAttrPutN

SubsetAttrPutN affecte une valeur numérique à un attribut d'un sous-ensemble spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetAttrPutN(NumericValue, DimensionName, SubName, AttrName, [LocalLangCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de dimension.
DimensionName	Dimension pour laquelle vous voulez affecter une valeur d'attribut.
SubName	Nom d'un sous-ensemble dans une dimension.
AttrName	Attribut dont vous voulez affecter la valeur.



Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 07161994 à l'attribut AccountingCode du sous-ensemble Z (de la dimension Models) pour le code d'environnement local de langue français.

```
SubsetAttrPutN(07161994, 'Models', 'Z', 'AccountingCode','fr');
```

### SubsetAttrInsert

SubsetAttrInsert crée un nouvel attribut pour les sous-ensembles de votre serveur TM1. Cette fonction crée un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de sous-ensemble existant, commencez par le supprimer à l'aide de la fonction SubsetAttrDelete. Vous pouvez ensuite utiliser SubsetAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

### Syntaxe

```
SubsetAttrInsert( Dimension, PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
Dimension	Nom de la dimension dont les sous-ensembles sont en cours de mise à jour.
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent ou que vous souhaitez que le nouvel attribut soit le premier attribut des sous-ensembles, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de sous-ensemble.

Argument	Description
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"> <li>• N - Crée un attribut numérique.</li> <li>• S - Crée un attribut chaîne.</li> <li>• A - Crée un attribut alias.</li> </ul>

### Exemple

Cet exemple crée l'attribut de chaîne Description pour les sous-ensembles de la dimension Model. Le paramètre PrevAttrName étant absent, cet attribut est inséré comme premier attribut des sous-ensembles sur votre serveur TM1.

```
SubsetAttrInsert('Model', '', 'Description', 'S');
```

### SubsetAttrDelete

SubsetAttrDelete supprime un attribut de sous-ensemble de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetAttrDelete(Dimension, AttrName);
```

Argument	Description
Dimension	Nom de la dimension dont l'attribut de sous-ensemble est en cours de suppression.
AttrName	Nom de l'attribut de dimension à supprimer.

### Exemple

Cet exemple supprime l'attribut Description des sous-ensembles de la dimension Model.

```
SubsetAttrDelete('Model', 'Description');
```

### ViewAttrDelete

ViewAttrDelete supprime un attribut de vue d'un cube spécifique de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewAttrDelete(CubeName, AttrName);
```

Argument	Description
CubeName	Nom du cube dont vous souhaitez supprimer l'attribut de vue.
AttrName	Nom de l'attribut de vue à supprimer.

## Exemple

Cet exemple supprime l'attribut Description des vues du cube Sales de votre serveur TM1.

```
ViewAttrDelete('Sales', 'Description');
```

## ViewAttrInsert

ViewAttrInsert crée un nouvel attribut pour les vues d'un cube spécifique de votre serveur TM1. Cette fonction peut créer un attribut chaîne, numérique ou alias.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



**Avertissement :** Pour mettre à jour un attribut de vue existant, commencez par le supprimer à l'aide de la fonction ViewAttrDelete. Vous pouvez ensuite utiliser ViewAttrInsert pour recréer l'attribut avec les modifications désirées.

**Important :** Si vous tentez de mettre à jour un attribut existant sans le supprimer au préalable, l'insertion échoue sans message d'erreur ou d'avertissement. L'attribut existant reste inchangé. Il n'est ni mis à jour, ni écrasé.

## Syntaxe

```
ViewAttrInsert( CubeName, PrevAttrName, NewAttrName, AttrType);
```

Argument	Description
CubeName	Cube parent pour lequel vous souhaitez insérer un attribut de vue.
PrevAttrName	Attribut précédant l'attribut que vous créez. S'il n'existe pas d'attribut précédent, ou si vous souhaitez que le nouvel attribut soit le premier attribut des vues, laissez cet argument vide.
NewAttrName	Nom à affecter au nouvel attribut de vue.
AttrType	Type d'attribut. Il existe trois valeurs possibles pour l'argument AttrType : <ul style="list-style-type: none"><li>• N - Crée un attribut numérique.</li><li>• S - Crée un attribut chaîne.</li><li>• A - Crée un attribut alias.</li></ul>

## Exemple

Cet exemple crée l'attribut de type chaîne Description pour les vues du cube Sales. Cet attribut est inséré à la suite de l'attribut Owner.

```
ViewAttrInsert('Sales', 'Owner', 'Description', 'S');
```

## ViewAttrN

ViewAttrN renvoie un attribut numérique d'une vue donnée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewAttrN(CubeName, ViewName, AttrName);
```

Argument	Description
CubeName	Nom de cube valide.
ViewName	Nom de vue valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la vue.

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut Category\_Code de la vue Product du cube Sales.

```
ViewAttrN('Sales', 'Product', 'Category_Code');
```

### ViewAttrNL

ViewAttrNL renvoie la valeur numérique d'un attribut pour une vue spécifiée, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewAttrNL(CubeName, ViewName, AttrName, [LangLocaleCode]);
```

Argument	Description
CubeName	Cube parent de la vue dont vous souhaitez extraire la valeur d'attribut.
ViewName	Nom de vue valide.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la vue.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur numérique de l'attribut `Category_Code` de la vue `Product` du cube `Sales` pour l'environnement local français.

```
ViewAttrNL('Sales', 'Product', 'Category_Code', 'fr');
```

### ViewAttrPutN

Cette fonction affecte une valeur à un attribut de vue numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewAttrPutN(NumericValue, CubeName, ViewName, AttrName, [LangLocaleCode] );
```

Argument	Description
NumericValue	Valeur à affecter à un attribut de vue.
CubeName	Cube parent de la vue pour laquelle vous voulez affecter une valeur d'attribut.

Argument	Description
ViewName	Vue à laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur 8222001 à l'attribut Category\_Code de la vue Product du cube Sales pour le code d'environnement local de langue français.

```
ViewAttrPutN(8222001, 'Sales', 'Product', 'Category_Code', 'fr');
```

### ViewAttrPutS

ViewAttrPutS affecte une valeur de chaîne à un attribut d'une vue spécifiée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewAttrPutS(String, CubeName, ViewName, AttrName, [LangLocaleCode] );
```

Argument	Description
String	Chaîne à affecter à un attribut de vue.
CubeName	Cube parent de la vue pour laquelle vous voulez affecter une valeur d'attribut.
ViewName	Nom de la vue pour laquelle vous voulez affecter une valeur d'attribut.
AttrName	Attribut dont vous voulez affecter la valeur.

Argument	Description
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur numérique s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux ISO 639-1 listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, la valeur de l'attribut de base est mise à jour.</p>

### Exemple

Cet exemple affecte la valeur de type chaîne Rocheford à l'attribut Owner de la vue Individual\_Stores du cube Sales pour le code d'environnement local de langue français.

```
ViewAttrPutS('Rocheford', 'Sales', 'Individual_Stores', 'Owner', 'fr');
```

## ViewAttrS

ViewAttrS renvoie un attribut de type chaîne pour une vue donnée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewAttrS(CubeName, ViewName, AttrName);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous voulez renvoyer une valeur d'attribut.
ViewName	Vue pour laquelle vous voulez renvoyer une valeur d'attribut.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la vue.

### Exemple

```
ViewAttrS('Plan_Business_Unit', 'Sales', 'Manager');
```

Dans cet exemple, la fonction renvoie la valeur de chaîne de l'attribut Manager de la vue Sales du cube Plan\_Business\_Unit.

## ViewAttrSL

ViewAttrSL renvoie la valeur de chaîne d'un attribut pour une vue spécifiée, en fonction d'un environnement local donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewAttrSL(CubeName, ViewName, AttrName, [LangLocaleCode]);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous voulez renvoyer une valeur d'attribut.
ViewName	Vue pour laquelle vous voulez renvoyer une valeur d'attribut.
AttrName	Attribut dont vous voulez obtenir la valeur. Cet argument doit être un attribut valide de la vue.
LangLocaleCode	<p>Ce paramètre facultatif indique le code d'environnement local de langue auquel la valeur s'applique.</p> <p>Les valeurs admises pour le code d'environnement local de langue correspondent aux codes de langue internationaux listés dans la dimension de contrôle Cultures.</p> <p>Lorsque le code d'environnement local de langue n'est pas défini ou lorsqu'il est omis, l'environnement local de l'utilisateur est utilisé comme argument LangLocaleCode.</p> <p>Si l'attribut n'a pas de valeur correspondant au code d'environnement local de langue, celle d'un code parent associé est renvoyée. S'il n'existe pas de valeur de l'attribut pour un code d'environnement local de langue parent associé, la valeur de l'attribut de base est renvoyée.</p> <p>Par exemple, si la valeur de LangLocaleCode est <code>fr-CA</code>, la fonction renvoie la valeur de l'attribut pour l'environnement local <code>fr-CA</code> (Français-Canada), si elle est disponible. Si la valeur de l'attribut n'est pas disponible pour <code>fr-CA</code>, la fonction tente de renvoyer la valeur de l'attribut de l'environnement local <code>fr</code> (français) parent. Si la valeur de l'attribut n'est pas disponible pour <code>fr</code>, l'attribut de base est renvoyé.</p>

### Exemple

Dans cet exemple, la fonction renvoie la valeur de type chaîne de l'attribut Manager de la vue Sales du cube Plan\_Business\_Unit pour l'environnement local français (Canada).

```
ViewAttrSL('Plan_Business_Unit', 'Sales', 'Manager', 'fr-CA');
```

## Fonctions TurboIntegrator de gestion de tâches

Ces fonctions sont utilisées pour la manipulation des tâches.



## ChoreError

La fonction `ChoreError` entraîne l'arrêt immédiat d'une tâche. Elle peut être appelée à partir de n'importe quel processus au sein d'une tâche. La fonction `ChoreError TurboIntegrator` entraîne l'arrêt immédiat d'une seule tâche. Les tâches arrêtées avec cette fonction sont signalées par un statut d'erreur.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

### Syntaxe

```
ChoreError;
```

### Arguments

Aucun.

## ChoreQuit

Cette fonction entraîne l'arrêt immédiat d'une tâche. Elle peut être appelée à partir de n'importe quel processus au sein d'une tâche. La tâche en cours est arrêtée et reçoit un statut d'erreur ; un message est généré pour indiquer que la fonction `ChoreQuit` a été appelée pour arrêter la tâche.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

### Syntaxe

```
ChoreQuit;
```

### Arguments

Aucun.

## ChoreRollback

`ChoreRollback` lance l'annulation d'une tâche. Si elle est utilisée dans un processus `TurboIntegrator`, cette fonction rejette toutes les éditions en attente et annule les traitements supplémentaires. Un message d'erreur apparaît dans les fichiers `tm1server.log` et `tm1processorerrorXXX.log`.

Si elle est utilisée dans une tâche en mode validation unique, la fonction **`ChoreRollback`** rejette toutes les éditions en attente de tous les processus précédents et l'exécution de la tâche s'arrête avec un code d'erreur. Si elle est utilisée dans une tâche en mode validations multiples, **`ChoreRollback`** rejette toutes les éditions en attente des processus en cours et l'exécution de la tâche s'arrête avec un code d'erreur. Les modifications déjà validées ne peuvent pas être annulées.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

### Syntaxe

```
ChoreRollback;
```

### Arguments

Aucun.

## SetChoreVerboseMessages

La fonction `SetChoreVerboseMessages` est utilisée pour activer (ou désactiver) la transmission de messages plus détaillés dans le fichier `Tm1s.log`. Vous pouvez utiliser cette fonction pour déboguer des tâches dans lesquelles plusieurs processus s'appellent les uns les autres avec la fonction `ExecuteProcess`.

Cette fonction est valide uniquement dans les processus `TM1 TurboIntegrator`.

## Syntaxe

```
SetChoreVerboseMessages(Flag);
```

La valeur zéro désactive la sortie de ces messages alors qu'une valeur autre que zéro active la sortie de messages plus détaillés. Par défaut, ce signal est désactivé.

Utilisez cette fonction pour activer (ou désactiver) des rapports plus complets sur les messages écrits dans le fichier Tm1s.log. Cette fonction est de la plus grande utilité dans le débogage des tâches dans lesquelles plusieurs processus s'appellent l'un l'autre à l'aide de la fonction ExecuteProcess.

La valeur zéro désactive la sortie de ces messages alors qu'une valeur autre que zéro active la sortie de messages plus détaillés. Par défaut, ce signal est désactivé.

Argument	Description
Flag	Réglez sur une valeur non nulle pour activer les messages plus détaillés. Réglez sur zéro (valeur par défaut) pour désactiver les messages plus détaillés.

## Fonctions TurboIntegrator de manipulation de cubes

Ces fonctions sont utilisées pour la manipulation des cubes.

### AddCubeDependency

AddCubeDependency permet de prédéfinir des interdépendances de cubes pour éviter des problèmes de conflit d'accès lors d'une utilisation normale du système.

Dans le cadre d'un fonctionnement normal, des dépendances de cubes sont établies lors de l'extraction de données dépassant les limites des cubes (données calculées par des règles qui font référence à un cube externe, par exemple). Pour créer les informations relatives aux dépendances, le serveur doit verrouiller les cubes au risque de bloquer l'accès aux cubes pendant toute la durée d'un calcul de vue. Puisqu'il s'agit d'un verrouillage en écriture, les utilisateurs ne peuvent plus accéder aux cubes. La fonction AddCubeDependency permet d'identifier les dépendances dès le démarrage du serveur, ce qui évite toute contention ultérieure du verrouillage puisqu'aucune nouvelle dépendance ne devra être établie.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
AddCubeDependency(BaseCube, DependentCube);
```

Argument	Description
BaseCube	Nom du cube dont dépend le cube dépendant.
DependentCube	Nom du cube qui dépend d'un autre cube (BaseCube) pour les données. Il s'agit généralement d'un cube qui utilise des règles pour extraire des données d'un cube externe.

### Exemple

```
Prenons l'exemple d'un cube nommé 'SalesCube' contenant la règle ['net']=!Units *  
DB('PriceCube' ... );
```

Dans cet exemple, 'SalesCube' est le cube dépendant, car il dépend des valeurs du cube de base nommé 'PriceCube' pour calculer la valeur de 'net'. Pour établir cette dépendance, vous devez exécuter la fonction suivante dans un processus TurboIntegrator : `AddCubeDependency( 'PriceCube' , 'SalesCube' );`

Pour établir une dépendance au moment du chargement du serveur, vous pouvez créer un processus qui exécute la fonction `AddCubeDependency`, planifier le processus en tant que tâche, puis définir cette tâche comme l'une des tâches **StartupChores** du fichier `Tm1s.cfg`.

## CellGetN

CellGetN extrait une valeur d'une cellule de cube numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellGetN(Cube, e1, e2 [...en]);
```

Argument	Description
Cube	Nom du cube où se trouve la valeur à récupérer.
e1,...en	<p>Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.</p> <p>Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.</p> <p>Si l'une des dimensions du cube utilise des hiérarchies, vous pouvez également utiliser la convention 'HierarchyName' : 'ElementName' pour spécifier un élément dans une hiérarchie spécifique. Dans ce cas, la séquence d'arguments doit toujours respecter l'ordre des dimensions dans le cube. Par exemple, si vous souhaitez faire référence à plusieurs éléments provenant de hiérarchies dans la deuxième dimension du cube, ces arguments doivent apparaître après l'argument destiné à la première dimension dans le cube et avant l'argument destiné à la troisième dimension dans le cube.</p>

### Exemple de CELLGETN et de variables

Lorsque cette fonction est utilisée dans une instruction conditionnelle (IF), l'instruction est la partie contenant la condition, pas l'ensemble du bloc conditionnel. En cas d'erreur mineure, l'exécution se poursuit avec l'instruction suivante. Une fois l'erreur mineure traitée, le traitement TurboIntegrator ne peut pas savoir qu'il s'agissait d'une instruction conditionnelle. Il considère donc la ligne suivante comme la prochaine instruction et non la ligne qui suit ENDIF.

Pour éviter cette situation, utilisez des variables pour chaque opération susceptible de rencontrer une erreur mineure, puis utilisez les variables dans l'instruction conditionnelle.

```
V1 = CELLGETN('PNLCube', 'fred', 'argentina', 'Sales', 'Jan'); IF(V1 = 454);
  ASCIIOUTPUT('bug.txt', 'if logic not working properly');
ENDIF;
```

### Exemple de CellGetN sans hiérarchies

Cet exemple présente une fonction qui utilise des noms d'élément simples comme arguments et ne fait pas référence à d'autres hiérarchies. Il récupère la valeur numérique à l'intersection des éléments Actual, Argentina, S Series 1.8L Sedan, Sales et Jan dans le cube Sales.

```
CellGetN ('Sales', 'Actual', 'Argentina', 'S Series1.8L Sedan', 'Sales', 'Jan');
```

### Exemple de CellGetN faisant référence à plusieurs hiérarchies

Cet exemple présente une fonction qui fait référence à plusieurs hiérarchies dans la dimension Model. Il récupère la valeur numérique à l'intersection des éléments Actual, Argentina, S Series (dans la hiérarchie vehicles de la dimension model), 2.8 Litre (dans la hiérarchie enginesize de la dimension model) et les éléments Jan du cube Sales.

```
CellGetN('Sales', 'Actual', 'Argentina',
('vehicles':'S Series', 'enginesize':'2.8 Litre'), 'Sales', 'Jan');
```

**Remarque :** Cet exemple fractionne artificiellement la ligne de code pour faciliter la lecture.

## CellGetS

CellGetS extrait une valeur d'une cellule de cube de chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellGetS(Cube, e1, e2 [,...en]);
```

Argument	Description
Cube	Nom du cube où se trouve la valeur à récupérer.
e1,...en	Noms des éléments de dimension définissant l'intersection du cube contenant la valeur à récupérer.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

Voir la note de la page «[CellGetN](#)», à la page 283 relative à l'application de la logique IF avec cette fonction.

### Exemple

```
CellGetS('Personnel', 'Commercial', 'Europe', 'Produit');
```

Cet exemple récupère la valeur de chaîne à l'intersection des éléments Commercial, Europe et Produit dans le cube Personnel.

## CellIncrementN

CellIncrementN incrémente une valeur de cellule numérique existante en appliquant la valeur indiquée. Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellIncrementN(x, Cube, e1, e2 [...en]);
```

Argument	Description
x	Valeur numérique à ajouter à une valeur de cellule existante.
Cube	Nom du cube auquel vous voulez envoyer la valeur.
e1,...en	Noms d'éléments de dimension définissant l'intersection de cube devant recevoir la valeur.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

```
CellIncrementN(1000, 'y2ksales', 'Actual', 'Argentina', 'S Series 1.8L Sedan', 'Sales', 'Jan');
```

Dans cet exemple, la valeur située à l'intersection des éléments Réel, Argentine, Berline 1,8L Série S, Ventes et Jan dans le cube ventesy2k est incrémentée de 1000.

## CellIsUpdateable

CellIsUpdateable détermine si une cellule de cube est accessible en écriture. Elle renvoie 1 s'il est possible d'ouvrir dans la cellule, sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellIsUpdateable(Cube, e1, e2 [...en]);
```

Argument	Description
Cube	Nom du cube où vous voulez écrire une valeur.

Argument	Description
$e_1, \dots, e_n$	Noms d'éléments de dimension définissant la cellule où vous voulez écrire une valeur.  Les arguments $e_1$ à $e_n$ doivent suivre un ordre précis. $e_1$ doit être un élément de la première dimension du cube, $e_2$ un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

```
CellIsUpdateable ('ventesy2k', 'Réal', 'Argentine', 'Berline 1,8L Série S', 'Ventes', 'Jan');
```

Cet exemple détermine s'il est possible d'écrire dans la cellule définie par les éléments Réel, Argentine, Berline Série S 1.8L, Ventes et Jan dans le cube ventes y2k. Si cette cellule peut recevoir une valeur, la fonction renvoie 1 ; sinon, elle renvoie 0.

### CellPutN

CellPutN envoie une valeur numérique à une cellule de cube.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellPutN(x, Cube, e1, e2 [, ..., en]);
```

Argument	Description
x	Valeur numérique.
Cube	Nom du cube auquel vous voulez envoyer la valeur.
$e_1, e_2, \dots, e_n$	Noms d'éléments de dimension définissant l'intersection de cube devant recevoir la valeur.  Les arguments $e_1$ à $e_n$ doivent suivre un ordre précis. $e_1$ doit être un élément de la première dimension du cube, $e_2$ un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

**Remarque :** Si vous fournissez des arguments non valides à la fonction CellPutN() dans un processus TurboIntegrator lorsque le cube n'existe pas, une erreur est envoyée à `tm1server.log`.

### Exemple

```
CellPutN (12345, 'ventesy2k', 'Réal', 'Argentine', 'Berline 1,8L Série S', 'Ventes', 'Jan');
```

Cet exemple envoie la valeur 12345 à l'intersection des éléments Réel, Argentine, Berline 1,8L Série S, Ventes et Jan dans le cube ventesy2k.

## CellPutProportionalSpread

CellPutProportionalSpread répartit une valeur spécifiée entre les feuilles d'une consolidation proportionnellement aux valeurs de cellule existantes. CellPutProportionalSpread remplace les valeurs de cellule existantes ; elle ne peut être utilisée pour ajouter et soustraire des valeurs de cellule existantes.

Cette fonction est analogue à la méthode de répartition de données dite Répartition proportionnelle qui est décrite en détail dans la documentation *TM1 Perspectives and TM1 Architect*. Si vous devez ajouter ou soustraire des valeurs de cellule existantes, utilisez la méthode de Répartition proportionnelle, qui peut être exécutée à l'aide de l'interface utilisateur ou d'une syntaxe de répartition de données.

**Remarque :** Lorsque vous utilisez CellPutProportionalSpread pour répartir une valeur dans les feuilles d'une consolidation, seules les feuilles contenant déjà des valeurs non nulles sont modifiées. La raison en est que les valeurs nulles ne peuvent pas être incrémentées ou décrémentées ; toute proportion de zéro reste zéro.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellPutProportionalSpread( value, cube, e1, e2, e3...,en );
```

Argument	Description
value	Valeur à répartir.
cube	Nom du cube dans lequel vous voulez répartir la valeur.
e1...en	Noms des éléments identifiant la consolidation dont les feuilles accepteront la valeur répartie.  Les arguments e1 à en doivent suivre un ordre précis. e1 doit être un élément de la première dimension du cube, e2 un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

```
CellPutProportionalSpread(7000,'SalesCube', 'Actual','North America',  
'S Series 1.8L Sedan', 'Sales', 'Jan')
```

Cet exemple distribue la valeur 7000 aux enfants de la consolidation SalesCube identifiée par les éléments Actual, North America, S Series 1,8L Sedan, Sales et Jan.

## CellPutS

CellPutS envoie une valeur de type chaîne à une cellule de cube.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellPutS(String, Cube, e1, e2 [...en]);
```

Argument	Description
String	Chaîne.
Cube	Nom du cube auquel vous voulez envoyer la chaîne.
<i>e1,...en</i>	Noms d'éléments de dimension définissant l'intersection du cube devant recevoir la chaîne.  Les arguments <i>e1</i> à <i>en</i> doivent suivre un ordre précis. <i>e1</i> doit être un élément de la première dimension du cube, <i>e2</i> un élément de la seconde, et ainsi de suite. Ces arguments peuvent aussi être les noms d'alias pour des éléments de dimension ou des variables TurboIntegrator.

### Exemple

```
CellPutS('Dupont', 'Personnel', 'Commercial', 'Europe', 'Produit');
```

Cet exemple envoie la valeur 'Dupont' à l'intersection des éléments Commercial, Europe et Produit dans le cube Personnel.

### CubeClearData

CubeClearData efface toutes les données d'un cube. Mieux vaut utiliser cette fonction plutôt que de réaliser une opération (création d'une vue qui couvre l'ensemble du cube) et d'effectuer un ViewZeroOut() pour effacer l'intégralité du cube ; c'est bien plus rapide.

Lorsque vous utilisez CubeClearData pour effacer les données d'un cube, toutes les cellules du cube contenant des feeders sont également effacées. Vous devez sauvegarder à nouveau la règle établissant les feeders ou restaurer le contenu de ces cellules via la fonction CubeProcessFeeders.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

**Remarque :** Cet appel supprime uniquement les données de cube, il ne supprime ni ne recrée le cube proprement dit. Il est important de retenir ce point en cas d'utilisation de bacs à sable. Si un cube est supprimé puis recréé, tous les bacs à sable qu'un utilisateur pourrait avoir seront abandonnés puisque le cube pour lequel ces bacs à sable ont été créés a été supprimé (et ce, même si un cube a été recréé avec le même nom). Si toutefois l'appel CubeClearData() est utilisé, les données des bacs à sable resteront valides puisque le cube pour lequel le bac à sable a été créé existe toujours.

### Syntaxe

```
CubeClearData( name-of-cube-as-string );
```

### Argument

Nom du cube à effacer, sous forme de chaîne.

### Exemple

```
CubeClearData( 'expense' );
```



## CubeCreate

CubeCreate crée un cube à partir de dimensions spécifiées. L'ordre des dimensions spécifié dans la fonction sera l'ordre des dimensions dans la définition de cube. Après son exécution, CubeCreate sauvegarde automatiquement le fichier .cub résultant sur disque.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeCreate(Cube, d1, d2 [...dn]);
```

Argument	Description
Cube	Nom à affecter au cube.
d1,...dn	Noms des dimensions constituant le cube. Vous devez spécifier au moins deux dimensions, mais pas plus de 256.

### Exemple

```
CubeCreate('Ventesy2k', 'Réelebud', 'Région', 'Modèle ', 'Compte1', 'Mois');
```

Cet exemple crée un cube nommé ventesy2k utilisant les dimensions Réelebud, Région, Modèle, Compte1 et Mois.

## CubeDestroy

CubeDestroy supprime le cube TM1 spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDestroy(Cube);
```

Argument	Description
Cube	Nom du cube à supprimer.

### Exemple

```
CubeDestroy('ventesy2k');
```

Cet exemple supprime le cube nommé y2ksales.

## CubeDimensionCountGet

CubeDimensionCountGet renvoie le nombre de dimensions dans un cube.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDimensionCountGet(CubeName);
```

Argument	Description
CubeName	Nom du cube dont vous souhaitez déterminer le nombre de dimensions.

### Exemple

```
CubeDimensionCountGet('Sales');
```

Dans cet exemple, la fonction renvoie le nombre de dimensions dans le cube Sales.

## CubeExists

CubeExists détermine si un cube spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. La fonction renvoie 1 si le cube existe, sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeExists(CubeName);
```

Argument	Description
CubeName	Nom du cube dont vous voulez confirmer l'existence.

### Exemple

```
CubeExists('Inventaire');
```

Cet exemple détermine si le cube Inventaire existe sur le serveur.

## CubeGetLogChanges

CubeGetLogChanges renvoie la valeur booléenne de la propriété Journalisation pour un cube spécifié.

La propriété Journalisation est définie dans la boîte de dialogue Affectations de sécurité TM1 et stockée dans le cube de contrôle }CubeProperties. Si la journalisation est activée pour un cube, la fonction renvoie 1. Si la journalisation est activée pour un cube, la fonction renvoie 1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeGetLogChanges(CubeName);
```

Argument	Description
CubeName	Cube dont vous voulez renvoyer la valeur de la propriété Journalisation.

### Exemple

```
CubeGetLogChanges('ventes2002');
```

Si la propriété Journalisation est activée pour le cube 2002sales, la fonction renvoie 1.

## CubeSaveData

CubeSaveData() sérialise un cube.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Dans un souci d'optimisation des performances, la journalisation des transactions peut être désactivée lors du chargement des données. Pour protéger les données nouvellement chargées dans l'éventualité d'une panne de serveur, il est possible de sérialiser les modifications sur le disque. La commande SaveDataAll a été utilisée pour sérialiser des données sur le disque et pour tronquer le journal des transactions. Lorsqu'il traite une commande SaveDataAll, le serveur acquiert un verrou READ sur chaque cube et un verrou IX sur chaque cube modifié. Si la commande SaveDataAll est exécutée pendant des périodes d'activité utilisateur, cela peut générer un conflit important avec cette activité.

En règle générale, tous les cubes affectés par SaveDataAll n'ont pas besoin d'être sérialisés car ils ne sont pas tous chargés avec de nouvelles données. La commande CubeSaveData permet de sérialiser un cube individuel sur le disque. Cette commande sérialise les données du cube qui ont été validées en mémoire, y compris les modifications qui leur ont été apportées dans le processus TurboIntegrator en cours mais qui n'ont pas encore validées.

### Syntaxe

```
CubeSaveData(Cube);
```

Argument	Description
Cube	Nom du cube à sérialiser.

### Exemple

```
CubeSaveData ('SalesCube');
```

Prenons en exemple le code du processus TurboIntegrator suivant :

```
CellPutN(500, 'y2ksales', 'Actual', 'Argentina', 'S Series 1.8 L Wagon', 'Sales', 'Jan');  
CubeSaveData('y2ksales');  
CellPutN(1000, 'y2ksales', 'Actual', 'Argentina', 'S Series 1.8 L Wagon', 'Sales', 'Jan');
```

Lors du traitement de la commande CubeSaveData, la valeur 500 correspondant à la cellule Ventes du mois de janvier est incluse dans la sérialisation du cube sur le disque, même si elle n'a pas encore été validée. En revanche, la mise à jour de la cellule Ventes du mois de janvier portée à 1 000 ne sera pas incluse dans la sérialisation.

### Journal des transactions

Une nouvelle entrée transactionnelle apparaît dans le journal des transactions lors de l'exécution de la commande CubeSaveData. Lorsqu'un fichier journal des transactions est traité au cours d'une reprise, toutes les mises à jour d'un cube appliquées jusqu'alors sont annulées lorsqu'une directive de CubeSaveData sur le cube est trouvée, puisque l'ensemble des mises à jour a déjà été sérialisé dans le cube.

### Reprise sur incident de serveur

La commande SaveDataAll tire parti du fait que tous les cubes sont verrouillés pendant son traitement, et elle tronque le journal des transactions car toutes les mises à jour effectuées avant la sérialisation ont été stockées sur le disque en toute sécurité. Comme ce n'est pas le cas pour la commande CubeSaveData, vous devez modifier la procédure de reprise des données en cas de sérialisation d'un cube.

Le fichier journal des transactions pourrait contenir des enregistrements représentant des modifications antérieures aux dernières données du cube et ne devant pas être appliqués lors de la reprise des données.

## CubeSetConnParams

CubeConnParams permet de chiffrer le mot de passe d'un cube virtuel dans le cube }CubeProperties.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeSetConnParams(cubeName, providerName, dataSourceLocation,dataSourceName,
dataSourceCatalog, userID, password, sapClientID, sapClientLang, providerString);
```

Argument	Description
cubeName	Nom du cube dont vous voulez définir le mot de passe.
providerName	
dataSourceLocation	Nom affecté par l'administrateur à un ensemble de catalogues à un emplacement particulier. Dans Microsoft Analysis Services, il s'agit du nom d'un serveur enregistré.
dataSourceName	
dataSource catalog	Nom affecté par votre administrateur à une collection particulière de bases de données (cubes, dimensions et autres objets). Pour MAS, il s'agit du nom de la base de données.
ID utilisateur	Nom d'utilisateur valide pour la base de données.
Password	Mot de passe à utiliser pour cette source de données.
sapClientID	ID client SAP
sapClientLang	Paramètre de langue de SAP.
providerString	

### Exemple

```
CubeSetConnParams(sc, TM10LAP, tm1server, , sdata, admin, apple, , , );
```

## CubeSetLogChanges

CubeSetLogChanges définit la propriété Journalisation d'un cube.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeSetLogChanges(Cube, LogChanges);
```

Argument	Description
Cube	Nom du cube pour lequel vous voulez définir la propriété LOGGING.
LogChanges	Valeur booléenne à affecter à la propriété. 1=Propriété LOGGING activée, 0=Propriété LOGGING désactivée.

## CubeTimeLastUpdated

EnableBulkLoadMode renvoie une valeur de série qui indique la date et l'heure de la dernière mise à jour d'un cube spécifié.

La valeur de série renvoyée par cette fonction utilise comme heure de début Jan 1 1900 12:00:00 A.M., ce qui correspond à la valeur 1.0. Les dates sont représentées par des entiers et les heures par des nombres décimaux compris entre .0 et .999999. Cette notation est conforme à la manière dont les valeurs de série de date et d'heure sont stockées et présentées dans Microsoft Excel.

**Remarque :** Par défaut, les valeurs de série de date et d'heure TM1 utilisent comme heure de début Jan 1 1960 12:00:00 A.M. Pour corriger l'incohérence de ces valeurs entre Excel et TM1, vous pouvez définir **UseExcelSerialDate=T** dans le fichier Tm1s.cfg pour indiquer au serveur TM1 d'utiliser des valeurs conformes aux normes Excel.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeTimeLastUpdated(cube);
```

Argument	Description
cube	Nom du cube.

### Exemple

```
CubeTimeLastUpdated('Sales');
```

Cet exemple renvoie une valeur correspondant à l'heure de la dernière mise à jour du cube Sales.

## CubeUnload

CubeUnload décharge de la mémoire le cube spécifié, ainsi que toutes les vues associées.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeUnload(CubeName);
```

Argument	Description
CubeName	Cube à décharger de la mémoire.

### Exemple

```
CubeUnload('Budgetfabrication');
```

Cet exemple décharge de la mémoire du serveur le cube ManufacturingBudget ainsi que toutes les vues qui lui sont associées.

## Fonctions TurboIntegrator de réservation de données

Utilisez les fonctions TurboIntegrator (TI) suivantes pour acquérir, libérer et gérer des réservations de données par programmation.

Pour plus de détails sur l'utilisation de la fonction Réservation de données, reportez-vous à la rubrique "Utilisation des réservations de données" dans la documentation IBM Cognos *TM1 for Developers*.

### CubeDataReservationAcquire

Cette fonction acquiert une réservation de données pour le cube, l'utilisateur et le tuple spécifiés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

`CubeDataReservationAcquire(Cube, User, bForce, Address, [AddressDelimiter])`

Argument	Description
Cube	Nom du cube.
User	Nom du propriétaire de la nouvelle réservation. Le nom d'utilisateur fourni sera validé afin de s'assurer qu'il s'agit d'un utilisateur existant.
bForce	Valeur booléenne qui détermine le comportement lorsque la réservation demandée entre en conflit avec une réservation existante. Si la valeur est égale à 0 (false), la demande est refusée s'il existe un conflit avec une réservation existante. Si la valeur est égale à 1 (true) et que l'utilisateur exécutant le processus TurboIntegrator dispose de la fonction DataReservationOverride, les réservations en conflit sont libérées et celle demandée est accordée.
Address	Séquence de chaîne segmentée des noms d'élément qui utilisent le tuple. L'ordre doit correspondre à l'ordre des dimensions d'origine du cube. Toutes les cellules du cube se trouvant dans le tuple constituent la région réservée. Vous pouvez choisir un élément dans chaque dimension ou utiliser une chaîne vide entre les délimiteurs pour sélectionner une dimension entière. En fonction de l'emplacement dans la hiérarchie de l'élément, la demande réserve une cellule, une tranche ou l'ensemble du cube.
AddressDelimiter	Chaîne de caractères facultatifs utilisée pour séparer les noms d'élément dans le paramètre d'adresse. La valeur par défaut est ' '.

#### Valeur renvoyée

Boolean - Renvoie true si l'acquisition a abouti.

## Exemple

```
CubeDataReservationAcquire('DRTestCube','User1',0,'ElemX|ElemY|ElemZ');
```

L'exemple suivant attribut la valeur 1 au paramètre bForce pour provoquer la demande de réservation de données lorsqu'il existe un conflit et utilise un caractère de délimiteur différent pour le paramètre AddressDelimiter.

```
CubeDataReservationAcquire('DRTestCube','User2',1,'ElemX*ElemY*ElemZ','*');
```

## CubeDataReservationRelease

Cette fonction libère la réservation de données spécifiée.

Si l'utilisateur spécifié n'est pas le propriétaire de la réservation, la libération aboutit uniquement si la fonction DataReservationOverride est activée pour l'utilisateur spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDataReservationRelease(Cube, User, Address, [AddressDelimiter])
```

Argument	Description
Cube	Nom du cube.
User	Nom du propriétaire de la réservation. Le nom d'utilisateur fourni sera validé afin de s'assurer qu'il s'agit d'un utilisateur existant.
Address	Séquence de chaîne segmentée des noms d'élément qui utilisent le tuple. L'ordre doit correspondre à l'ordre des dimensions d'origine du cube.
AddressDelimiter	Chaîne de caractères facultatifs utilisée pour séparer les noms d'élément dans le paramètre d'adresse. La valeur par défaut est ' '.

### Valeur renvoyée

Boolean - Renvoie true si l'acquisition a abouti.

### Exemple

```
CubeDataReservationRelease('DRTestCube','User1','ElemX|ElemY|ElemZ');
```

L'exemple suivant utilise un autre caractère pour le paramètre AddressDelimiter.

```
CubeDataReservationRelease('DRTestCube','User2','ElemX*ElemY*ElemZ','*');
```

## CubeDataReservationReleaseAll

Cette fonction libère plusieurs réservations de données existantes.

Toutes les réservations se trouvant dans l'adresse spécifiée qui correspondent au filtre utilisateur sont libérées. Un filtre utilisateur vide correspond à tous les utilisateurs.

Si le filtre utilisateur spécifié ne correspond pas à l'utilisateur exécutant le processus TurboIntegrator, la fonction DataReservationOverride doit être activée.





Argument	Description
User	Nom du propriétaire de réservation à utiliser en tant que filtre. Si cette zone est laissée vide, la fonction renvoie des réservations pour tout propriétaire. Lorsqu'un nom est indiqué, la fonction filtre les résultats uniquement pour le propriétaire spécifié.
AddressDelimiter	Chaîne de caractères facultatifs utilisée pour séparer les noms d'élément dans le paramètre d'adresse renvoyé. La valeur par défaut est ' '.

### Valeur renvoyée

Address - Heure de la création de la réservation, nom du propriétaire de la réservation et adresse de la réservation. Les différents éléments s'affichent dans l'ordre suivant : heure de création, délimiteur, ID utilisateur, délimiteur, ID d'élément séparés par le délimiteur dans l'ordre des dimensions du cube (ordre d'origine).

Une chaîne vide s'affiche lorsqu'il n'existe aucune entrée pour l'index spécifié.

Le format de la valeur affichée est :

```
[creation time][delimiter][owner name][delimiter][element1][delimiter]
[element2][delimiter]...[elementN]
```

Par exemple :

```
"20100622211601|Fred Bloggs|Element1|Element2|Element3"
```

**Remarque :** Les réservations peuvent changer lors de la répétition de la liste des réservations.

L'utilisation de l'index ne garantit pas l'obtention d'une liste complète de réservations. Des réservations peuvent être ajoutées ou retirées à tout emplacement de la liste, des réservations peuvent donc être ignorées ou répétées lors d'une boucle dans les valeurs d'index.

Si le filtre du propriétaire est spécifié, l'index s'applique uniquement aux membres de la liste filtrée. Si les propriétaires sont indiqués dans l'ordre suivant dans la liste des réservations, User1, User1, User2, et que la demande spécifie un propriétaire User2, un index 1 extrait le troisième membre de la liste.

### Exemple

```
CubeDataReservationGet(1,'DRTestCube','User1','*');
```

```
CubeDataReservationGet(1,'DRTestCube','');
```

L'exemple suivant trouve toutes les réservations appartenant à l'utilisateur Fred Bloggs dans le cube d'entrée Expense et les utilise :

```
vIndex = 1;
vCube = 'Expense Input';
vUserFilter = 'Fred Bloggs';
vDelim = '|';
vAddress = CubeDataReservationGet( vIndex, vCube, vUserFilter,vDelim);
WHILE (vAddress @<> '');
    vSep1 = SCAN( vDelim, vAddress);
    vDRUser = SUBST( vAddress, 1, vSep1 - 1);
    vDRAddress = SUBST( vAddress, vSep1 + 1, LONG(vDRAddress) - vSep1);

#    do something meaningful with the
#    user and reservation address here
    vIndex = vIndex + 1;
```

```
vAddress = CubeDataReservationGet( vIndex, vCube, vUserFilter,vDelim);  
END;
```

## CubeDataReservationGetConflicts

Cette fonction recherche les réservations existantes sur un cube spécifique pouvant être à l'origine de conflits avec le tuple, l'adresse et l'utilisateur spécifiés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDataReservationGetConflicts(Index, Cube, User,  
Address, [AddressDelimiter]) returns ConflictAddress;
```

Argument	Description
Index	Index à boucle unique à utiliser pour la répétition dans les conflits qui répondent à cette requête.
Cube	Nom du cube à rechercher
User	La requête recherche des réservations entrant en conflit avec cet utilisateur.
Address	Séquence de chaîne segmentée des noms d'élément qui utilisent le tuple. L'ordre doit correspondre à l'ordre des dimensions d'origine du cube.
AddressDelimiter	Chaîne de caractères facultatifs utilisée pour séparer les noms d'élément dans le paramètre d'adresse.  La valeur par défaut est ' '.

### Valeur renvoyée

ConflictAddress - Heure de la création de la réservation, nom du propriétaire de la réservation et adresse de la réservation. Les différents éléments s'affichent dans l'ordre suivant : heure de création, délimiteur, ID utilisateur, délimiteur, ID d'élément séparés par le délimiteur dans l'ordre des dimensions du cube (ordre d'origine).

Une chaîne vide s'affiche lorsqu'il n'existe aucune entrée pour l'index spécifié.

Le format de la valeur affichée est :

```
[creation time][delimiter][owner name][delimiter][element1][delimiter]  
[element2][delimiter]...[elementN]
```

Par exemple :

```
"20100622211601|Fred Bloggs|Element1|Element2|Element3"
```

**Remarque :** Les réservations peuvent changer lors de la répétition de la liste des réservations de conflit. L'utilisation de l'index ne garantit pas l'obtention d'une liste complète de réservations. Des réservations peuvent être ajoutées ou retirées à tout emplacement de la liste, des réservations peuvent donc être ignorées ou répétées lors d'une boucle dans les valeurs d'index.

## CubeDRAcquire

Cette fonction acquiert une réservation de données pour le cube, l'utilisateur et le tuple spécifiés. Alors que la fonction `CubeDataReservationAcquire` s'applique aux dimensions ne comportant qu'une seule hiérarchie, cette fonction s'applique aux dimensions en contenant plusieurs.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDRAcquire(Cube, User, bForce, Element-list)
```

Argument	Description
Cube	Nom du cube.
User	Nom du propriétaire de la nouvelle réservation. Le nom d'utilisateur fourni sera validé afin de s'assurer qu'il s'agit d'un utilisateur existant.
bForce	Valeur booléenne qui détermine le comportement lorsque la réservation demandée entre en conflit avec une réservation existante. Si la valeur est égale à 0 (false), la demande est refusée s'il existe un conflit avec une réservation existante. Si la valeur est égale à 1 (true) et que l'utilisateur exécutant le processus TurboIntegrator dispose de la fonction <code>DataReservationOverride</code> , les réservations en conflit sont libérées et celle demandée est accordée.
Element-list	Les coordonnées sont exprimées sous forme d'une liste de longueur variable d'arguments de nom d'élément individuels. Chacun des paramètres de nom d'élément doit être qualifié par la hiérarchie. Si la dimension ne comporte qu'une seule hiérarchie, utilisez le nom de la dimension. Sinon, utilisez le nom de la hiérarchie prévue. Les éléments sont spécifiés dans le format '<hiérarchie>;!<élément>', chaque instance étant séparée par une virgule.

### Valeur renvoyée

Boolean - Renvoie true si l'acquisition a abouti.

### Exemple

```
CubeDRAcquire('DRTestCube','User1',0,'Hier1':'ElemX','Hier2':'ElemY','Hier3':'ElemZ');
```

L'exemple suivant attribut la valeur 1 au paramètre `bForce` pour provoquer la demande de réservation de données lorsqu'il existe un conflit et utilise un caractère de délimiteur différent pour le paramètre `AddressDelimiter`.

```
CubeDRAcquire('DRTestCube','User2',1,'Hier1':'ElemX','Hier2':'ElemY','Hier3':'ElemZ');
```

## CubeDRRelease

Cette fonction libère la réservation de données spécifiée. Alors que la fonction `CubeDataReservationRelease` s'applique aux dimensions ne comportant qu'une seule hiérarchie, cette fonction s'applique aux dimensions en contenant plusieurs.

Si l'utilisateur spécifié n'est pas le propriétaire de la réservation, la libération aboutit uniquement si la fonction `DataReservationOverride` est activée pour l'utilisateur spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDRRelease(Cube, User, Element-list)
```

Argument	Description
Cube	Nom du cube.
User	Nom du propriétaire de la réservation. Le nom d'utilisateur fourni sera validé afin de s'assurer qu'il s'agit d'un utilisateur existant.
Element-list	Les coordonnées sont exprimées sous forme d'une liste de longueur variable d'arguments de nom d'élément individuels. Chacun des paramètres de nom d'élément doit être qualifié par la hiérarchie. Si la dimension ne comporte qu'une seule hiérarchie, utilisez le nom de la dimension. Sinon, utilisez le nom de la hiérarchie prévue. Les éléments sont spécifiés dans le format '<hiérarchie>;'<élément>', chaque instance étant séparée par une virgule.

### Valeur renvoyée

Boolean - Renvoie true si l'acquisition a abouti.

### Exemple

```
CubeDRRelease('DRTestCube', 'User1', 'Hier1': 'ElemX', 'Hier2': 'ElemY', 'Hier3': 'ElemZ');
```

L'exemple suivant utilise un autre caractère pour le paramètre `AddressDelimiter`.

```
CubeDRRelease('DRTestCube', 'User2', 'Hier1': 'ElemX', 'Hier2': 'ElemY', 'Hier3': 'ElemZ');
```

## CubeDRReleaseAll

Cette fonction libère plusieurs réservations de données existantes. Alors que la fonction `CubeDataReservationReleaseAll` s'applique aux dimensions ne comportant qu'une seule hiérarchie, cette fonction s'applique aux dimensions en contenant plusieurs.

Toutes les réservations se trouvant dans l'adresse spécifiée qui correspondent au filtre utilisateur sont libérées. Un filtre utilisateur vide correspond à tous les utilisateurs.

Si le filtre utilisateur spécifié ne correspond pas à l'utilisateur exécutant le processus TurboIntegrator, la fonction `DataReservationOverride` doit être activée.

L'utilisation d'un filtre utilisateur vide et de caractères génériques dans la zone d'adresse libère toutes les réservations.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeDRReleaseAll(Cube, UserFilter, Element-list)
```

Argument	Description
Cube	Nom du cube.
UserFilter	Filtre de nom d'utilisateur devant correspondre aux réservations existantes.
Element-list	Les coordonnées sont exprimées sous forme d'une liste de longueur variable d'arguments de nom d'élément individuels. Chacun des paramètres de nom d'élément doit être qualifié par la hiérarchie. Si la dimension ne comporte qu'une seule hiérarchie, utilisez le nom de la dimension. Sinon, utilisez le nom de la hiérarchie prévue.  Les éléments sont spécifiés dans le format '<hiérarchie>;': '<élément>', chaque instance étant séparée par une virgule.

## Valeur renvoyée

Boolean - Renvoie true lorsqu'aucune erreur ne survient.

## Exemple

```
CubeDRReleaseAll('DRTestCube', 'User1', 'Hier1': 'ElemX', 'Hier2': 'ElemY', 'Hier3': 'ElemZ');
```

L'exemple suivant libère toutes les réservations dans le cube spécifié pour tous les utilisateurs.

```
CubeDRReleaseAll('DRTestCube', '', '');
```

## CubeDRGet

Cette fonction recherche les réservations existantes sur un cube spécifique pour un utilisateur ou tous les utilisateurs. Alors que la fonction `CubeDataReservationGet` s'applique aux dimensions ne comportant qu'une seule hiérarchie, cette fonction s'applique aux dimensions en contenant plusieurs.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeDRGet(Index, Cube, User, Element-list) returns Address;
```

Argument	Description
Index	Index à boucle unique à utiliser pour la répétition dans les réservations sur le cube spécifié.
Cube	Nom du cube à rechercher.
User	Nom du propriétaire de réservation à utiliser en tant que filtre.  Si cette zone est laissée vide, la fonction renvoie des réservations pour tout propriétaire.  Lorsqu'un nom est indiqué, la fonction filtre les résultats uniquement pour le propriétaire spécifié.

Argument	Description
Element-list	<p>Les coordonnées sont exprimées sous forme d'une liste de longueur variable d'arguments de nom d'élément individuels. Chacun des paramètres de nom d'élément doit être qualifié par la hiérarchie. Si la dimension ne comporte qu'une seule hiérarchie, utilisez le nom de la dimension. Sinon, utilisez le nom de la hiérarchie prévue.</p> <p>Les éléments sont spécifiés dans le format '&lt;hiérarchie&gt;;!&lt;élément&gt;', chaque instance étant séparée par une virgule.</p>

### Valeur renvoyée

Address - Heure de la création de la réservation, nom du propriétaire de la réservation et adresse de la réservation. Les différents éléments s'affichent dans l'ordre suivant : heure de création, délimiteur, ID utilisateur, délimiteur, ID d'élément séparés par le délimiteur dans l'ordre des dimensions du cube (ordre d'origine).

Une chaîne vide s'affiche lorsqu'il n'existe aucune entrée pour l'index spécifié.

Le format de la valeur affichée est :

```
[creation time][delimiter][owner name][delimiter][element1][delimiter]
[element2][delimiter]...[elementN]
```

Par exemple :

```
"20100622211601|Fred Bloggs|Element1|Element2|Element3"
```

**Remarque :** Les réservations peuvent changer lors de la répétition de la liste des réservations.

L'utilisation de l'index ne garantit pas l'obtention d'une liste complète de réservations. Des réservations peuvent être ajoutées ou retirées à tout emplacement de la liste, des réservations peuvent donc être ignorées ou répétées lors d'une boucle dans les valeurs d'index.

Si le filtre du propriétaire est spécifié, l'index s'applique uniquement aux membres de la liste filtrée. Si les propriétaires sont indiqués dans l'ordre suivant dans la liste des réservations, User1, User1, User2, et que la demande spécifie un propriétaire User2, un index 1 extrait le troisième membre de la liste.

### Exemple

```
CubeDRGet(1,'DRTestCube','User1','*');
```

```
CubeDRGet(1,'DRTestCube','');
```

L'exemple suivant trouve toutes les réservations appartenant à l'utilisateur Fred Bloggs dans le cube d'entrée Expense et les utilise :

```
vIndex = 1;
vCube = 'Expense Input';
vUserFilter = 'Fred Bloggs';
vHier = 'Currency';
vElem = 'Local Currency';
vAddress = CubeDRGet( vIndex, vCube, vUserFilter, vHier:vElem);
WHILE (vAddress @<> '');
    vSep1 = SCAN( vHier:vElem, vAddress);
    vDRUser = SUBST( vAddress, 1, vSep1 - 1);
    vDRAddress = SUBST( vAddress, vSep1 + 1, LONG(vDRAddress) - vSep1);

    # do something meaningful with the
    user and reservation address here
    vIndex = vIndex + 1;
    vAddress = CubeDRGet( vIndex, vCube, vUserFilter, vHier:vElem);
END;
```

## CubeDRGetConflicts

Cette fonction recherche les réservations existantes sur un cube spécifique pouvant être à l'origine de conflits avec l'utilisateur spécifié. Alors que la fonction CubeDataReservationGetConflicts s'applique aux dimensions ne comportant qu'une seule hiérarchie, cette fonction s'applique aux dimensions en contenant plusieurs.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CubeDRGetConflicts(Index, Cube, User,  
Element-list) returns ConflictAddress;
```

Argument	Description
Index	Index à boucle unique à utiliser pour la répétition dans les conflits qui répondent à cette requête.
Cube	Nom du cube à rechercher
User	La requête recherche des réservations entrant en conflit avec cet utilisateur.
Element-list	Les coordonnées sont exprimées sous forme d'une liste de longueur variable d'arguments de nom d'élément individuels. Chacun des paramètres de nom d'élément doit être qualifié par la hiérarchie. Si la dimension ne comporte qu'une seule hiérarchie, utilisez le nom de la dimension. Sinon, utilisez le nom de la hiérarchie prévue.  Les éléments sont spécifiés dans le format '<hiérarchie>;'<élément>', chaque instance étant séparée par une virgule.

### Valeur renvoyée

ConflictAddress - Heure de la création de la réservation, nom du propriétaire de la réservation et adresse de la réservation. Les différents éléments s'affichent dans l'ordre suivant : heure de création, délimiteur, ID utilisateur, délimiteur, ID d'élément séparés par le délimiteur dans l'ordre des dimensions du cube (ordre d'origine).

Une chaîne vide s'affiche lorsqu'il n'existe aucune entrée pour l'index spécifié.

Le format de la valeur affichée est :

```
[creation time][delimiter][owner name][delimiter][element1][delimiter]  
[element2][delimiter]...[elementN]
```

Par exemple :

```
"20100622211601|Fred Bloggs|Element1|Element2|Element3"
```

**Remarque :** Les réservations peuvent changer lors de la répétition de la liste des réservations de conflit. L'utilisation de l'index ne garantit pas l'obtention d'une liste complète de réservations. Des réservations peuvent être ajoutées ou retirées à tout emplacement de la liste, des réservations peuvent donc être ignorées ou répétées lors d'une boucle dans les valeurs d'index.

## Fonctions TurboIntegrator de date et d'heure

Ces fonctions permettent la mise en forme et l'analyse des dates et des heures selon une grande variété de formats et de paramètres régionaux.

### FormatDate

FormatDate met en forme une valeur de date selon un formateur défini à l'aide de la fonction NewDateFormatter.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

```
FormatDate(Date, <Pattern>, <Index>)
```

Argument	Description
<i>Date</i>	Valeur de date. Le type de valeur, série ou UNIX, doit correspondre au formateur utilisé.
<i>Pattern</i>	Masque utilisé pour le formatage de dates. Pour obtenir la liste complète des syntaxes de format, accédez à l'adresse suivante : <a href="http://userguide.icu-project.org/formatparse/datetime">http://userguide.icu-project.org/formatparse/datetime</a> . Si une chaîne vide est utilisée, le format est déterminé par l'environnement local en fonction des paramètres FormatterStyle et FormatterType spécifiés avec la fonction NewDateFormatter.
<i>Index</i>	Index renvoyé par un appel à la fonction NewDateFormatter. La valeur par défaut est 0. Si aucun formateur de date n'existe à l'index, le formateur par défaut est utilisé comme s'il avait été créé avec l'appel suivant : NewDateFormatter(' ', 'Etc/UTC', 'serial', 'medium', 'date')

#### Exemple

```
sDate = FormatDate(18000);
```

### NewDateFormatter

NewDateFormatter définit un formateur de date. Elle renvoie un index qui sera utilisé dans les fonctions ParseDate et FormatDate. Les indices commencent à 0 et sont incrémentés de 1 lors de chaque appel à NewDateFormat. Les formateurs de date sont valides au cours de l'exécution du processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

```
NewDateFormatter(Locale, <TimeZone>, <UseUNIXTime>, <FormatterStyle>, <FormatterType>, <TimeType>)
```



Argument	Description
<i>Locale</i>	Paramètre régional utilisé pour le formatage et l'analyse des dates. Si une chaîne vide est transmise, les paramètres régionaux du système d'exploitation sont utilisés. Les paramètres régionaux sont spécifiés dans le format langue[_pays][.variant]. Par exemple, cs_CK correspond à la langue tchèque et à la république tchèque.
<i>TimeZone</i>	Fuseau horaire utilisé pour l'analyse et le formatage des dates. Pour obtenir la liste complète des fuseaux horaires, accédez à l'adresse suivante : <a href="http://en.wikipedia.org/wiki/List_of_tz_database_time_zones">http://en.wikipedia.org/wiki/List_of_tz_database_time_zones</a> . Si aucun fuseau horaire n'est spécifié, la valeur utilisée est UTC ( 'Etc/UTC' ).
<i>UseUNIXTime</i>	Si 'unix' est spécifié, les heures sont traitées, en millisecondes, à partir du 1er janvier 1970. Sinon, elles sont traitées selon le format série TM1. Notez que seules les dates postérieures au 1er janvier 1970 peuvent être traitées, même si le format série TM1 est utilisé.
<i>FormatterStyle</i>	Permet de contrôler le format de date utilisé lorsqu'un masque vide est spécifié dans une fonction FormatDate ou ParseDate. Les valeurs valides sont 'full', 'long', 'medium' et 'short'. La valeur par défaut est 'medium'.
<i>FormatterType</i>	Permet de contrôler le type de format utilisé lorsqu'un masque vide est spécifié dans une fonction FormatDate ou ParseDate. Les valeurs valides sont 'time', 'date' et 'datetime'. La valeur par défaut est 'date'.

### Exemple

```
dfUNIX = NewDateFormatter('', 'Etc/UTC', 'unix');
```

```
dfStyleFullDateTime = NewDateFormatter('en_us', 'America/Toronto', 'serial', 'full', 'datetime');
```

### ParseDate

ParseDate analyse une chaîne de date en fonction d'un formateur défini avec la fonction NewDateFormatter.

Une valeur de date est renvoyée, série ou UNIX selon le formateur spécifié. Si la date ne peut pas être analysée, une valeur indéfinie est renvoyée. Peut être testée avec la fonction ISUND.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ParseDate (DateString, <Pattern>, <Index>)
```

Argument	Description
<i>DateString</i>	Chaîne de date.

Argument	Description
<i>Pattern</i>	Masque utilisé pour l'analyse de dates. Pour obtenir la liste complète des syntaxes de format, accédez à l'adresse suivante : <a href="http://userguide.icu-project.org/formatparse/datetime">http://userguide.icu-project.org/formatparse/datetime</a> . Si une chaîne vide est utilisée, le format est déterminé par l'environnement local en fonction des paramètres <code>FormatterStyle</code> et <code>FormatterType</code> spécifiés avec la fonction <code>NewDateFormatter</code> .
<i>Index</i>	Index renvoyé par un appel à la fonction <code>NewDateFormatter</code> . La valeur par défaut est 0. Si aucun formateur de date n'existe à l'index, le formateur par défaut est utilisé comme s'il avait été créé avec l'appel suivant : <code>NewDateFormatter('', 'Etc/UTC', 'serial', 'medium', 'date')</code>

### Exemple

```
nDate = ParseDate('2011/11/24', 'yyyy/MM/dd');
```

## Fonctions TurboIntegrator de manipulation de dimensions

Ces fonctions facilitent la manipulation des attributs.

### DimensionCreate

Cette fonction crée une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

```
DimensionCreate(DimName);
```

Argument	Description
DimName	Nom à affecter à la dimension.

### Exemple

```
DimensionCreate('Produit');
```

Cet exemple crée la dimension `Produit`.

### DimensionDeleteAllElements

`DimensionDeleteAllElements` supprime tous les éléments d'une dimension. Cette fonction est utile pour recréer des hiérarchies de dimension.

**Remarque :** La suppression d'un élément supprime toutes les données de cube identifiées par cet élément. Toutefois, si vous utilisez `DimensionDeleteAllElements` pour supprimer des éléments, puis recréez ces éléments avec le même nom dans l'onglet `Métadonnées`, toute cellule de données d'un cube identifiée par les éléments sera conservée après la régénération de la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionDeleteAllElements(DimName);
```

Argument	Description
DimName	Nom de la dimension dont vous voulez supprimer tous les éléments.

## Exemple

```
DimensionDeleteAllElements('Modèle');
```

Cette fonction supprime tous les éléments de la dimension Modèle.

## DimensionDeleteElements

DimensionDeleteElements supprime tous les éléments d'une dimension à l'aide du sous-ensemble d'éléments. Tous les éléments du sous-ensemble référencé sont supprimés, y compris les éléments de niveau C.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionDeleteElements (DimensionName, Subset)
```

Argument	Description
DimensionName	Nom de la dimension dont vous voulez supprimer le sous-ensemble d'éléments.
Subset	Liste des éléments à supprimer de la dimension indiquée. Le sous-ensemble est généralement temporaire.

## DimensionDestroy

DimensionDestroy supprime une dimension de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionDestroy(DimName);
```

Argument	Description
DimName	Nom de la dimension à supprimer.

## Exemple

```
DimensionDestroy('Product');
```

Dans cet exemple, la dimension Product est supprimée de la base de données TM1.

## DimensionElementComponentAdd

DimensionElementComponentAdd ajoute un composant (enfant) à un élément consolidé. Vous ne pouvez pas utiliser cette fonction dans la procédure Epilogue d'un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionElementComponentAdd(DimName, ConsolidatedElName, ElName, ElWeight);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé auquel vous voulez ajouter un enfant.
ConsolidatedElName	Élément auquel vous voulez ajouter un enfant.
ElName	Nom de l'élément enfant.
ElWeight	Pondération de l'élément enfant.

### Exemple

```
DimensionElementComponentAdd('Measures', 'Net Sales', 'Expenses', -1);
```

Cet exemple ajoute l'enfant Dépenses à la consolidation Ventes nettes dans la dimension Measures. L'enfant a une pondération de -1 dans la consolidation.

## DimensionElementComponentAddDirect

Cette fonction ajoute un composant (enfant) à un élément consolidé en éditant directement une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionElementComponentAdd, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
DimensionElementComponentAddDirect(DimName, ConsolidatedElName, ElName, ElWeight);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé auquel vous voulez ajouter un enfant.
ConsolidatedElName	Élément consolidé auquel vous voulez ajouter un enfant.
ElName	Nom de l'élément enfant.
ElWeight	Pondération de l'élément enfant.

## Exemple

```
DimensionElementComponentAddDirect('Measures', 'Net Sales', 'Expenses', -1);
```

Cet exemple ajoute l'enfant Expenses à la consolidation Net Sales dans la dimension Measures. L'enfant a une pondération de -1 dans la consolidation.

## DimensionElementComponentDelete

DimensionElementComponentDelete supprime un composant (enfant) d'un élément consolidé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionElementComponentDelete(DimName, ConsolidatedElName, ElName);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé dont vous voulez supprimer un enfant.
ConsolidatedElName	Élément consolidé dont vous voulez supprimer un enfant.
ElName	Nom de l'élément enfant à supprimer.

## Exemple

```
DimensionElementComponentDelete('Région', 'Benelux', 'Belgique');
```

Cet exemple supprime l'enfant Belgium de la consolidation Benelux dans la dimension Region.

## DimensionElementComponentDeleteDirect

DimensionElementComponentDeleteDirect supprime un composant (enfant) dans un élément consolidé en éditant directement la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et

finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionElementComponentDelete, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
DimensionElementComponentDeleteDirect(DimName, ConsolidatedElName, ElName);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé dont vous voulez supprimer un enfant.
ConsolidatedElName	Élément consolidé dont vous voulez supprimer un enfant.
ElName	Nom de l'élément enfant à supprimer.

## Exemple

```
DimensionElementComponentDeleteDirect('Region', 'Benelux', 'Belgium');
```

Cet exemple supprime l'enfant Belgium de la consolidation Benelux dans la dimension Region.

## DimensionElementDelete

DimensionElementDelete supprime un élément d'une dimension.

**Remarque :** La suppression d'un élément supprime toutes les données de cube identifiées par cet élément.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionElementDelete(DimName, ElName);
```

Argument	Description
DimName	Dimension contenant l'élément à supprimer.
ElName	Élément à supprimer.

## Exemple

```
DimensionElementDelete ('Region', 'Belgium');
```

Cet exemple supprime l'élément Belgium de la dimension Region.

## DimensionElementDeleteDirect

DimensionElementDeleteDirect supprime un élément dans une dimension en éditant directement la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

**Remarque :** La suppression d'un élément supprime toutes les données de cube identifiées par cet élément.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle.

TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionElementDelete, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
DimensionElementDeleteDirect(DimName, ElName);
```

Argument	Description
DimName	Dimension contenant l'élément à supprimer.
ElName	Élément à supprimer.

## Exemple

```
DimensionElementDeleteDirect('Region', 'Belgium');
```

Cet exemple supprime l'élément Belgium de la dimension Region.

## DimensionElementExists

DimensionElementExists détermine si un élément spécifique existe dans une dimension sur le serveur à partir duquel un processus TurboIntegrator est exécuté. La fonction renvoie 1 si les éléments existent dans la dimension sur le serveur. Sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionElementExists(DimName, ElName);
```

Argument	Description
DimName	Dimension contenant l'élément à rechercher. La dimension doit exister sur le serveur où le processus TurboIntegrator est exécuté.
ElName	Élément à rechercher.

## Exemple

Cet exemple détermine si l'élément Belgium existe dans la dimension Region sur le serveur.

```
DimensionElementExists('Region', 'Belgium');
```

## DimensionElementInsert

DimensionElementInsert ajoute un élément à une dimension. Vous pouvez utiliser cette fonction pour ajouter des éléments numériques, des éléments chaîne ou des éléments consolidés. Vous ne pouvez pas utiliser cette fonction dans la procédure Données ou Epilogue d'un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DimensionElementInsert(DimName, InsertionPoint, ElName, ElType);
```

Argument	Description
DimName	Dimension à laquelle vous voulez ajouter un élément.
InsertionPoint	Élément de dimension existant L'élément ajouté à la dimension sera inséré immédiatement avant cet élément existant. Si ce paramètre est une chaîne vide ("), le nouvel élément est ajouté à la fin de la dimension.
ElName	Nom que vous souhaitez attribuer au nouvel élément.
ElType	Type d'élément. Il existe trois valeurs ElType possibles : N - Signifie un élément numérique. S - Signifie un élément chaîne. C - Signifie un élément consolidé.

## Exemple

```
DimensionElementInsert('Region', 'Belgium', 'Netherlands', 'S');
```



Cet exemple ajoute l'élément de type chaîne Netherlands à la dimension Region. Netherlands est ajouté immédiatement avant Belgium dans la dimension.

### Exemple avec un point d'insertion vide

```
DimensionElementInsert('Region', '', 'Netherlands', 'S');
```

Cet exemple ajoute l'élément de type chaîne Netherlands à la dimension Region. Netherlands est ajouté à la fin de la dimension.

## DimensionElementInsertDirect

DimensionElementInsertDirect ajoute un élément à une dimension en éditant directement la dimension. Vous pouvez utiliser cette fonction pour ajouter des éléments numériques, des éléments chaîne ou des éléments consolidés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, la méthode d'édition d'une dimension dans TM1 consiste à utiliser un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionElementInsert, par exemple) sont utilisées dans l'onglet Métadonnées du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute insertion d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
DimensionElementInsertDirect(DimName, InsertionPoint, ElName, ElType);
```

Argument	Description
DimName	Dimension à laquelle vous voulez ajouter un nouvel élément.
InsertionPoint	Élément de dimension existant L'élément ajouté à la dimension sera inséré immédiatement avant cet élément existant. Si ce paramètre est vide, le nouvel élément est ajouté à la fin de la dimension.  Notez que cette fonction est optimisée pour le cas où le point d'insertion est transmis sous la forme d'une chaîne vide.
ElName	Nom à affecter au nouvel élément.

Argument	Description
ElType	Type d'élément. Il existe trois valeurs ElType possibles : N - Signifie un élément numérique. S - Signifie un élément chaîne. C - Signifie un élément consolidé.

### Exemple

```
DimensionElementInsertDirect('Region', 'Belgium', 'Netherlands', 'N');
```

Cet exemple ajoute l'élément numérique Netherlands à la dimension Region. Netherlands apparaît immédiatement avant Belgium dans la définition de dimension.

### DimensionElementPrincipalName

DimensionElementPrincipalName renvoie le nom principal d'un élément ou d'un alias d'élément.

TurboIntegrator doit utiliser les noms d'éléments principaux lors de la mise à jour de dimensions ; les alias d'éléments ne doivent pas être utilisés. Cette fonction est utile pour déterminer les noms d'éléments principaux lorsque vous essayez de mettre à jour une dimension et que seuls des alias d'élément sont disponibles pour le processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionElementPrincipalName( DimName, ElName )
```

Argument	Description
DimName	Nom de la dimension depuis laquelle vous voulez récupérer un nom d'élément principal.
ElName	Nom d'élément. ElName peut être un alias d'élément ou un nom d'élément principal.

### Exemple

Si ElName n'est pas dans la version enregistrée en cours de DimName, la fonction renvoie ElName.

Si ElName est dans DimName, que ce soit un alias d'élément ou un nom d'élément principal, la fonction renvoie le nom d'élément principal.

### DimensionExists

DimensionExists détermine si une dimension spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. La fonction renvoie 1 si la dimension existe, sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionExists(DimName);
```

Argument	Description
DimName	Nom de la dimension dont vous voulez confirmer l'existence.

### Exemple

```
DimensionExists('Région');
```

Cet exemple détermine si la dimension Region existe sur le serveur.

## DimensionHierarchyCreate

DimensionHierarchyCreate crée une hiérarchie dans une dimension existante. Le nom de la hiérarchie ne doit pas être le même que celui de la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionHierarchyCreate(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension existante qui inclura la hiérarchie.
HierName	Nom à affecter à la hiérarchie. Vous ne pouvez pas utiliser le nom de la dimension.

### Exemple

```
DimensionHierarchyCreate('Vehicles', 'Trucks');
```

Cet exemple crée une hiérarchie Trucks vide dans la dimension Vehicles.

## DimensionSortOrder

DimensionSortOrder définit un type de tri et un sens pour les éléments de dimension et les composants des éléments consolidés dans une dimension. L'ordre de tri défini par DimensionSortOrder détermine la manière dont le sous-ensemble s'affiche dans l'éditeur de sous-ensemble.

DimensionSortOrder définit les propriétés d'une dimension ; la dimension elle-même n'est pas triée jusqu'à ce qu'elle soit enregistrée sur le serveur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionSortOrder(DimName, CompSortType, CompSortSense, ElSortType, ElSortSense);
```

Argument	Description
DimName	Nom de la dimension pour laquelle vous voulez définir un ordre de tri.

Argument	Description
CompSortType	<p>Définit comment les composants des éléments consolidés apparaissent dans la dimension. Il existe deux valeurs CompSortType :</p> <p>ByInput - Garde l'ordre dans lequel les éléments ont été insérés dans les consolidations.</p> <p>ByName - Trie les composants des consolidations par nom.</p>
CompSortSense	<p>Définit le sens du tri pour les composants des consolidations. Cet argument est obligatoire mais il ne s'applique que quand le CompSortType est ByName. Il existe deux valeurs CompSortSense possibles :</p> <p>Ascending - Trie les composants des consolidations par ordre alphabétique croissant.</p> <p>Descending - Trie les composants des consolidations par ordre alphabétique décroissant.</p>
ElSortType	<p>Définit un ordre de tri pour les éléments de dimension. Il existe quatre valeurs ElSortType possibles :</p> <p>ByInput - Garde l'ordre dans lequel les éléments ont été insérés dans les dimensions.</p> <p>ByName - Trie les éléments de dimension par nom.</p> <p>ByLevel - Trie les éléments de dimension par niveau.</p> <p>ByHierarchy - Trie les éléments de dimension par hiérarchie.</p>
ElSortSense	<p>Définit le sens du tri pour les éléments de dimension. Cet argument est obligatoire mais il ne s'applique que quand le ElSortType est ByName ou ByLevel. Il existe deux valeurs ElSortSense possibles :</p> <p>Ascending - Trie les éléments de dimension par ordre croissant, soit alphabétiquement, soit par niveau.</p> <p>Descending - Trie les éléments de dimension par ordre décroissant, soit alphabétiquement, soit par niveau.</p>

### Exemple

```
DimensionSortOrder ('Region', 'ByName', 'Descending', 'ByLevel', 'Ascending');
```

Cet exemple définit un ordre de tri pour la dimension Region. Tous les éléments de dimension sont triés par ordre de niveau croissant et les composants de consolidation sont triés par ordre alphabétique décroissant.

## DimensionTimeLastUpdated

ElementSecurityGet renvoie une valeur de série qui indique la date et l'heure de la dernière mise à jour d'une dimension spécifiée.

La valeur de série renvoyée par cette fonction utilise comme heure de début Jan 1 1900 12:00:00 A.M., ce qui correspond à la valeur 1.0. Les dates sont représentées par des entiers et les heures par des nombres décimaux compris entre .0 et .999999. Cela est conforme à la manière dont les valeurs de série de date/heure sont stockées et signalées dans Microsoft Excel.

**Remarque :** Par défaut, les valeurs de série de date/heure TM1 utilisent comme heure de début Jan 1 1960 12:00:00 A.M. Pour résoudre l'incohérence de ces valeurs entre Excel et TM1, vous pouvez définir UseExcelSerialDate=T dans le fichier Tm1s.cfg pour indiquer au serveur TM1 d'utiliser des valeurs conformes aux normes Excel.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionTimeLastUpdated(dimension);
```

Argument	Description
dimension	Nom de la dimension.

### Exemple

```
DimensionTimeLastUpdated('Region');
```

Cet exemple renvoie des informations sur la date et l'heure de la dernière mise à jour de la dimension Region.

## DimensionTopElementInsert

DimensionTopElementInsert crée un élément racine dans une dimension. Si la dimension a déjà une racine unique, l'élément n'est pas créé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionTopElementInsert(DimName, InsertionPoint, ElName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un élément racine.
InsertionPoint	Élément de dimension existant L'élément racine ajouté à la dimension sera inséré immédiatement avant cet élément existant.
ElName	Nom à affecter au nouvel élément racine.

### Exemple

```
DimensionTopElementInsert('Region', 'Netherlands', 'World');
```

Cet exemple ajoute l'élément racine World à la dimension Region. World apparaît immédiatement avant Netherlands dans la définition de dimension.

## DimensionTopElementInsertDirect

DimensionTopElementInsertDirect crée un élément racine dans une dimension en éditant directement la dimension. Si la dimension a déjà une racine unique, l'élément n'est pas créé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle.

TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionTopElementInsert, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

### Syntaxe

```
DimensionTopElementInsertDirect(DimName, InsertionPoint, ElName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un élément racine.
InsertionPoint	Élément de dimension existant L'élément racine ajouté à la dimension sera inséré immédiatement avant cet élément existant.
ElName	Nom à affecter au nouvel élément racine.

### Exemple

```
DimensionTopElementInsertDirect('Region', 'Netherlands', 'World');
```

Cet exemple ajoute l'élément racine World à la dimension Region. World apparaît immédiatement avant Netherlands dans la définition de dimension.

## DimensionUpdateDirect

La fonction DimensionUpdateDirect effectue une réécriture complète d'une dimension qui a fait l'objet d'une édition directe dans un processus TurboIntegrator en compressant principalement l'encombrement de mémoire de la dimension.

Une dimension qui subit une série d'édérations directes (notamment des suppressions d'élément) finit par utiliser plus de mémoire que sa contrepartie entièrement réécrite ne l'aurait fait. Cette fonction facultative peut être utilisée après l'édition directe d'une dimension avec DimensionElementInsertDirect, DimensionElementDeleteDirect, DimensionElementComponentAddDirect, DimensionElementComponentDeleteDirect et/ou DimensionTopElementInsertDirect. L'appel à DimensionUpdateDirect induit un coût initial en termes de mémoire, dû à la copie complète qui doit être effectuée. Il garantit cependant que l'encombrement de la dimension est à son minimum à la fin du traitement.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DimensionUpdateDirect(DimName);
```

Argument	Description
DimName	Nom de la dimension à réécrire.

### Exemple

```
DimensionUpdateDirect('Region');
```

Dans cet exemple, la dimension Region est réécrite.

## Fonctions TurboIntegrator de manipulation des hiérarchies

Ces fonctions facilitent la manipulation des hiérarchies.

### HierarchyContainsAllLeaves

HierarchyContainsAllLeaves ne renvoie true que si la hiérarchie spécifiée contient l'intégralité des éléments feuilles présents dans la dimension. A savoir, elle contient tous les éléments feuille visibles dans la hiérarchie *Leaves* spéciale. S'il manque un ou plusieurs éléments feuille dans la hiérarchie spécifiée, cette fonction renvoie false.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyContainsAllLeaves(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension qui contient la hiérarchie de toutes les feuilles.
HierName	Nom de la hiérarchie que vous déterminez comme une hiérarchie de toutes les feuilles.

## Exemple

```
HierarchyContainsAllLeaves('Region', 'Leaves');
```

Cet exemple détermine si la hiérarchie Leaves de la dimension Region contient tous les membres feuille.

## HierarchyCreate

HierarchyCreate crée une hiérarchie dans une dimension existante. Le nom de la hiérarchie ne doit pas être le même que celui de la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyCreate(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension existante qui inclura la hiérarchie.
HierName	Nom à affecter à la hiérarchie. Vous ne pouvez pas utiliser le nom de la dimension.

## Exemple

```
HierarchyCreate('Vehicles', 'Trucks');
```

Cet exemple crée une hiérarchie Trucks vide dans la dimension Vehicles.

## HierarchyDeleteAllElements

HierarchyDeleteAllElements supprime tous les éléments d'une hiérarchie. Cette fonction est utile pour recréer des hiérarchies de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyDeleteAllElements(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension dont vous voulez supprimer tous les éléments.
HierName	Nom de la hiérarchie dans la dimension.

## Exemple

```
HierarchyDeleteAllElements('Equipment', 'Helmets');
```

Cet exemple supprime tous les éléments de la hiérarchie Helmets dans la dimension Equipment.



## HierarchyDeleteElements

HierarchyDeleteElements supprime les éléments feuilles d'une hiérarchie à l'aide d'un sous-ensemble d'éléments.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyDeleteElements (DimensionName, HierarchyName, Subset)
```

Argument	Description
DimensionName	Nom de la dimension dont vous voulez supprimer le sous-ensemble d'éléments.
HierarchyName	Nom de la hiérarchie dont vous voulez supprimer le sous-ensemble d'éléments.  Si la hiérarchie indiquée est Feuilles, le sous-ensemble doit répertorier les feuilles à supprimer, et ces dernières sont alors supprimées complètement de la dimension.
Subset	Liste des éléments à supprimer de la hiérarchie indiquée. Le sous-ensemble est généralement temporaire.

## HierarchyDestroy

HierarchyDestroy supprime une hiérarchie de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyDestroy(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension qui contient la hiérarchie.
HierName	Nom de la hiérarchie à supprimer.

### Exemple

```
HierarchyDestroy('Product', 'Transmissions');
```

Dans cet exemple, la hiérarchie Transmissions est supprimée de la base de données TM1.

## HierarchyElementComponentAdd

HierarchyElementComponentAdd ajoute un composant (enfant) à un élément consolidé. Vous ne pouvez pas utiliser cette fonction dans la procédure Epilogue d'un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyElementComponentAdd(DimName, HierName, ConsolidatedElName, ElName, ElWeight);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé auquel vous voulez ajouter un enfant.
HierName	Hierarchie de la dimension spécifiée.
ConsolidatedElName	Élément auquel vous voulez ajouter un enfant.
ElName	Nom de l'élément enfant.
ElWeight	Pondération de l'élément enfant.

### Exemple

```
HierarchyElementComponentAdd('Measures', 'Europe', 'Net Sales', 'Expenses', -1);
```

Cet exemple ajoute l'enfant Expenses à la consolidation Net Sales dans la hiérarchie Europe de la dimension Measures. L'enfant a une pondération de -1 dans la consolidation.

## HierarchyElementComponentAddDirect

HierarchyElementComponentAddDirect ajoute un composant (enfant) à un élément consolidé en éditant directement une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans Cognos TM1 est effectué en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (HierarchyElementComponentAdd, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
HierarchyElementComponentAddDirect(DimName, HierName, ConsolidatedElName, ElName, ElWeight);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé auquel vous voulez ajouter un enfant.
HierName	Hierarchie de la dimension spécifiée.
ConsolidatedElName	Élément consolidé auquel vous voulez ajouter un enfant.
ElName	Nom de l'élément enfant.
ElWeight	Pondération de l'élément enfant.

### Exemple

```
HierarchyElementComponentAddDirect('Measures', 'Europe', 'Net Sales',
'Expenses', -1);
```

Cet exemple ajoute l'enfant Expenses à la consolidation Net Sales dans la hiérarchie Europe de la dimension Measures. L'enfant a une pondération de -1 dans la consolidation.

## HierarchyElementComponentDelete

HierarchyElementComponentDelete supprime un composant (enfant) d'un élément consolidé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyElementComponentDelete(DimName, HierName, ConsolidatedElName, ElName);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé dont vous voulez supprimer un enfant.
HierName	Nom de la hiérarchie dans la dimension.
ConsolidatedElName	Élément consolidé dont vous voulez supprimer un enfant.
ElName	Nom de l'élément enfant à supprimer.

### Exemple

```
HierarchyElementComponentDelete('Region', 'Western', 'Benelux', 'Belgium');
```

Cet exemple supprime l'enfant Belgium de la consolidation Benelux dans la hiérarchie Western de la dimension Region.

## HierarchyElementComponentDeleteDirect

HierarchyElementComponentDeleteDirect supprime un composant (enfant) dans un élément consolidé en éditant directement la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, la méthode d'édition d'une dimension dans TM1 consiste à utiliser un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (HierarchyElementComponentDelete, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
HierarchyElementComponentDeleteDirect(DimName, HierName, ConsolidatedElName, ElName);
```

Argument	Description
DimName	Dimension parent de l'élément consolidé dont vous voulez supprimer un enfant.
HierName	Nom de la hiérarchie dans la dimension.
ConsolidatedElName	Élément consolidé dont vous voulez supprimer un enfant.
ElName	Nom de l'élément enfant à supprimer.

## Exemple

```
HierarchyElementComponentDeleteDirect('Region', 'Western', 'Benelux', 'Belgium');
```

Cet exemple supprime l'enfant Belgium de la consolidation Benelux dans la hiérarchie Western de la dimension Region.

## HierarchyElementDelete

HierarchyElementDelete supprime un élément d'une hiérarchie.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyElementDelete(DimName, HierName, ElName);
```

Argument	Description
DimName	Dimension contenant l'élément à supprimer.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément à supprimer de la hiérarchie.

### Exemple

```
HierarchyElementDelete('Region', 'Western', 'Belgium');
```

Cet exemple supprime l'élément Belgium de la hiérarchie Western dans la dimension Region.

### HierarchyElementDeleteDirect

HierarchyElementDeleteDirect supprime un élément d'une dimension en éditant directement la dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

**Remarque :** La suppression d'un élément supprime toutes les données de cube identifiées par cet élément.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (DimensionElementDelete, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

### Syntaxe

```
HierarchyElementDeleteDirect(DimName, HierName, ElName);
```

Argument	Description
DimName	Dimension contenant l'élément à supprimer.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément à supprimer.

## Exemple

```
HierarchyElementDeleteDirect('Region', 'Western', 'Belgium');
```

Cet exemple supprime l'élément Belgium de la hiérarchie Western dans la dimension Region.

## HierarchyElementExists

HierarchyElementExists détermine si un élément spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. La fonction renvoie 1 si les éléments existent dans la hiérarchie sur le serveur. Sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyElementExists(DimName, HierName, ElemName);
```

Argument	Description
DimName	Nom de la dimension contenant l'élément dont vous voulez vérifier l'existence.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément à rechercher dans la hiérarchie.

## Exemple

```
HierarchyElementExists('Region', 'Western', 'Belgium');
```

Cet exemple détermine si l'élément Belgium de la hiérarchie Western dans la dimension Region existe sur le serveur.

## HierarchyElementInsert

HierarchyElementInsert ajoute un élément à une dimension. Vous pouvez utiliser cette fonction pour ajouter des éléments numériques, des éléments chaîne ou des éléments consolidés. Vous ne pouvez pas utiliser cette fonction dans la procédure Données ou Epilogue d'un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyElementInsert(DimName, HierName, InsertionPoint, ElName, ElType);
```

Argument	Description
DimName	Dimension à laquelle vous voulez ajouter un nouvel élément.
HierName	Nom de la hiérarchie dans la dimension.
InsertionPoint	Élément de dimension existant L'élément ajouté à la dimension sera inséré immédiatement avant cet élément existant. Si ce paramètre est vide, le nouvel élément est ajouté à la fin de la dimension.

Argument	Description
ElName	Nom à affecter au nouvel élément.
ElType	Type d'élément. Il existe trois valeurs ElType possibles : N - Signifie un élément numérique. S - Signifie un élément chaîne. C - Signifie un élément consolidé.

### Exemple

```
HierarchyElementInsert('Region', 'Western', 'Belgium', 'Netherlands', 'N');
```

Cet exemple ajoute l'élément numérique Netherlands à la hiérarchie Western de la dimension Region. Netherlands apparaît immédiatement avant Belgium dans la définition de dimension.

### HierarchyElementInsertDirect

HierarchyElementInsertDirect ajoute un élément à une dimension en éditant directement la dimension. Vous pouvez utiliser cette fonction pour ajouter des éléments numériques, des éléments chaîne ou des éléments consolidés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle. TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (HierarchyElementInsert, par exemple) sont utilisées dans l'onglet Métadonnées du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute insertion d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

### Syntaxe

```
HierarchyElementInsertDirect(DimName, HierName, InsertionPoint, ElName, ElType);
```

Argument	Description
DimName	Dimension à laquelle vous voulez ajouter un nouvel élément.
HierName	Nom de la hiérarchie dans la dimension.

Argument	Description
InsertionPoint	Elément de dimension existant L'élément ajouté à la dimension sera inséré immédiatement avant cet élément existant. Si ce paramètre est vide, le nouvel élément est ajouté à la fin de la dimension.  Notez que cette fonction est optimisée pour le cas où le point d'insertion est transmis sous la forme d'une chaîne vide.
ElName	Nom à affecter au nouvel élément.
ElType	Type d'élément. Il existe trois valeurs ElType possibles :  N - Signifie un élément numérique. S - Signifie un élément chaîne. C - Signifie un élément consolidé.

### Exemple

```
HierarchyElementInsertDirect('Region', 'Western', 'Belgium', 'Netherlands','N');
```

Cet exemple ajoute l'élément numérique Netherlands à la hiérarchie Western de la dimension Region. Netherlands apparaît immédiatement avant Belgium dans la définition de dimension.

## HierarchyElementPrincipalName

HierarchyElementPrincipalName renvoie le nom principal d'un élément ou d'un alias d'élément.

TurboIntegrator doit utiliser les noms d'éléments principaux lors de la mise à jour de dimensions ; les alias d'éléments ne doivent pas être utilisés. Cette fonction est donc utile pour déterminer les noms d'éléments principaux lorsque vous essayez de mettre à jour une dimension et que seuls les alias d'éléments sont disponibles pour le processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyElementPrincipalName( DimName, HierName, ElName )
```

Argument	Description
DimName	Nom de la dimension depuis laquelle vous voulez récupérer un nom d'élément principal.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Nom d'élément. ElName peut être un alias d'élément ou un nom d'élément principal.

### Exemple

Si ElName n'est pas dans la version enregistrée en cours de DimName, la fonction renvoie ElName.

Si ElName est dans DimName, que ce soit un alias d'élément ou un nom d'élément principal, la fonction renvoie le nom d'élément principal.



## HierarchyExists

HierarchyExists détermine si une hiérarchie spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. La fonction renvoie 1 si la hiérarchie existe ; sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyExists(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension qui contient la hiérarchie dont vous souhaitez confirmer l'existence.
HierName	Nom de la hiérarchie dans la dimension.

### Exemple

```
HierarchyExists('Region', 'Europe');
```

Cet exemple détermine si la hiérarchie Europe de la dimension Region existe sur le serveur.

## HierarchyHasOrphanedLeaves

HierarchyHasOrphanedLeaves renvoie la valeur 1 si un ou plusieurs éléments de la hiérarchie spécifiée ne sont pas des composants d'un élément parent dans cette hiérarchie.

La fonction renvoie 1 si la hiérarchie existe sur le serveur ; sinon, elle renvoie 0. Elle renvoie 0 si tous les éléments feuilles de la hiérarchie sont un composant d'un ou plusieurs éléments parent. Les valeurs stockées pour ce type d'élément ne sont pas agrégées.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyHasOrphanedLeaves(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension qui contient la hiérarchie en cours de vérification.
HierName	Nom de la hiérarchie dans laquelle vous recherchez des membres feuille orphelins.

### Exemple

```
HierarchyHasOrphanedLeaves('Region', 'Europe');
```

Cet exemple détermine si la hiérarchie Europe de la dimension Region contient des feuilles orphelines.

## HierarchySortOrder

HierarchySortOrder définit un type de tri et un sens pour les éléments de dimension et les composants des éléments consolidés dans une dimension. L'ordre de tri défini par DimensionSortOrder détermine la manière dont le sous-ensemble s'affiche dans l'éditeur de sous-ensemble.

DimensionSortOrder définit les propriétés d'une dimension ; la dimension elle-même n'est pas triée jusqu'à ce qu'elle soit enregistrée sur le serveur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySortOrder(DimName, HierName, CompSortType, CompSortSense, ElSortType, ElSortSense);
```

Argument	Description
DimName	Nom de la dimension pour laquelle vous voulez définir un ordre de tri.
HierName	Nom de la hiérarchie dans la dimension.
CompSortType	Définit comment les composants des éléments consolidés apparaissent dans la dimension. Il existe deux valeurs CompSortType :  ByInput - Garde l'ordre dans lequel les éléments ont été insérés dans les consolidations.  ByName - Trie les composants des consolidations par nom.
CompSortSense	Définit le sens du tri pour les composants des consolidations. Cet argument est obligatoire mais il ne s'applique que quand le CompSortType est ByName. Il existe deux valeurs CompSortSense possibles :  Ascending - Trie les composants des consolidations par ordre alphabétique croissant.  Descending - Trie les composants des consolidations par ordre alphabétique décroissant.
ElSortType	Définit un ordre de tri pour les éléments de dimension. Il existe quatre valeurs ElSortType possibles :  ByInput - Garde l'ordre dans lequel les éléments ont été insérés dans les dimensions.  ByName - Trie les éléments de dimension par nom.  ByLevel - Trie les éléments de dimension par niveau.  ByHierarchy - Trie les éléments de dimension par hiérarchie.

Argument	Description
ElSortSense	Définit le sens du tri pour les éléments de dimension. Cet argument est obligatoire mais il ne s'applique que quand le ElSortType est ByName ou ByLevel. Il existe deux valeurs ElSortSense possibles :  Ascending - Trie les éléments de dimension par ordre croissant, soit alphabétiquement, soit par niveau.  Descending - Trie les éléments de dimension par ordre décroissant, soit alphabétiquement, soit par niveau.

### Exemple

```
HierarchySortOrder ('Region', 'Europe', 'ByName', 'Descending', 'ByLevel', 'Ascending');
```

Cet exemple définit un ordre de tri pour la hiérarchie Europe de la dimension Region. Tous les éléments de dimension sont triés par ordre de niveau croissant et les composants de consolidation sont triés par ordre alphabétique décroissant.

## HierarchyTimeLastUpdated

HierarchyTimeLastUpdated indique quand une hiérarchie de dimension spécifiée a été mise à jour pour la dernière fois. La fonction renvoie un nombre réel qui représente le jour en cours (avec l'heure, les minutes, les secondes et les millisecondes) depuis le début de l'année 1900.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyTimeLastUpdated(dimension, hierarchy);
```

Argument	Description
dimension	Nom de la dimension.
hierarchy	Nom de la hiérarchie.

### Exemple

```
HierarchyTimeLastUpdated('Region', 'Europe');
```

Cet exemple renvoie des informations sur la date et l'heure de la dernière mise à jour de la hiérarchie Europe de la dimension Region. Si la valeur 42548.<heures>.<minutes>.<millisecondes> est renvoyée, vous pouvez diviser 42548 par 365 pour obtenir (environ) 116. Lorsqu'il est ajouté à la date de début de 1900, le résultat correspond à l'année en cours, soit 2016.

## HierarchyTopElementInsert

HierarchyTopElementInsert crée un élément racine dans une dimension. Si la dimension a déjà une racine unique, l'élément n'est pas créé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyTopElementInsert(DimName, HierName, InsertionPoint, ElName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un élément racine.
HierName	Nom de la hiérarchie dans la dimension.
InsertionPoint	Élément de dimension existant L'élément racine ajouté à la dimension sera inséré immédiatement avant cet élément existant.
ElName	Nom à affecter au nouvel élément racine.

## Exemple

```
HierarchyTopElementInsert('Region', 'Western', 'Netherlands', 'World');
```

Cet exemple ajoute l'élément racine World à la hiérarchie Western de la dimension Region. World apparaît immédiatement avant Netherlands dans la définition de dimension.

## HierarchyTopElementInsertDirect

HierarchyTopElementInsertDirect crée un élément racine dans une dimension en éditant directement la dimension. Si la dimension a déjà une racine unique, l'élément n'est pas créé.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, l'édition d'une dimension dans TM1 se fait en utilisant un modèle d'édition de copie complète. Dans ce modèle, une copie d'édition de la dimension est créée, les modifications lui sont appliquées, et finalement, la dimension elle-même est réécrite en utilisant la copie d'édition comme modèle.

TurboIntegrator prend automatiquement en charge l'édition de copie complète lorsque les fonctions d'édition de dimension de TurboIntegrator (HierarchyTopElementInsert, par exemple) sont utilisées dans la procédure Metadata du processus. TurboIntegrator crée automatiquement la copie d'édition et lui applique les opérations d'édition, puis réécrit la dimension elle-même à la fin de la procédure Metadata.

L'édition directe a ceci de différent qu'aucune copie n'est utilisée. Les opérations sont effectuées directement sur la dimension elle-même. Il existe deux cas d'utilisation spécialisés auxquels l'édition directe est destinée :

- Lorsque l'objectif du processus TurboIntegrator est d'effectuer une petite modification dans une grande dimension. Dans ce cas, l'édition directe est plus rentable, car elle évite de copier et de réécrire entièrement une dimension de grande taille.
- Lorsque l'objectif du processus TurboIntegrator est de charger de gros volumes de données dans un cube. Dans ce cas, la procédure Metadata du processus est laissée vide à dessein, et toute modification d'élément nécessaire pour le support du chargement des données est effectuée à l'aide d'appels directs à la procédure Data. Lorsque la procédure Metadata est vide, le processus ignore une itération complète de la source de données externe, ce qui peut accélérer le chargement des données.

## Syntaxe

```
HierarchyTopElementInsertDirect(DimName, HierName, InsertionPoint, ElName);
```

Argument	Description
DimName	Dimension pour laquelle vous voulez créer un élément racine.
HierName	Nom de la hiérarchie dans la dimension.
InsertionPoint	Élément de dimension existant L'élément racine ajouté à la dimension sera inséré immédiatement avant cet élément existant.
ElName	Nom à affecter au nouvel élément racine.

### Exemple

```
HierarchyTopElementInsertDirect('Region', 'Western', 'Netherlands', 'World');
```

Cet exemple ajoute l'élément racine World à la hiérarchie Western de la dimension Region. World apparaît immédiatement avant Netherlands dans la définition de dimension.

### HierarchyUpdateDirect

HierarchyUpdateDirect effectue une réécriture complète d'une hiérarchie qui a fait l'objet d'une édition directe dans un processus TurboIntegrator en compressant principalement l'encombrement de mémoire de la hiérarchie.

Une dimension qui subit une série d'édérations directes (notamment des suppressions d'élément) finit par utiliser plus de mémoire que sa contrepartie entièrement réécrite ne l'aurait fait. Cette fonction peut éventuellement être utilisée après avoir directement édité une dimension avec HierarchyElementInsertDirect, HierarchyElementDeleteDirect, HierarchyElementComponentAddDirect, HierarchyElementComponentDeleteDirect et/ou HierarchyTopElementInsertDirect. L'appel de HierarchyUpdateDirect induit un coût initial en termes de mémoire, dû à la copie complète qui doit être effectuée. Il garantit cependant que l'encombrement de la dimension est à son minimum à la fin du traitement.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyUpdateDirect(DimName, HierName);
```

Argument	Description
DimName	Nom de la dimension à réécrire.
HierName	Nom de la hiérarchie dans la dimension.

### Exemple

```
HierarchyUpdateDirect('Region', 'Western');
```

Cet exemple réécrit la hiérarchie Western de la dimension Region.

## Fonctions TurboIntegrator ODBC

Ces fonctions facilitent la manipulation ODBC.

## ODBCClose

ODBCClose ferme une connexion à une source de données ODBC.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ODBCClose(Source);
```

Argument	Description
Source	Nom d'une source de données ODBC ouverte.

### Exemple

```
ODBCClose('Comptabilité');
```

Cette fonction ferme la connexion à la source de données ODBC Comptabilité.

## ODBCOpen

ODBCOpen ouvre une source de données ODBC pour la sortie.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ODBCOpen(Source, ClientName, Password);
```

Argument	Description
Source	Nom d'une source de données ODBC.
ClientName	Client valide sur la source de données.
Password	Mot de passe pour ClientName.

### Exemple

```
ODBCOpen('Comptabilité', 'Jleclerc', 'Poisson');
```

Cet exemple ouvre la source de données ODBC pour le client Jleclerc avec le mot de passe Poisson.

## ODBCOPENEx

ODBCOPENEx ouvre une source de données ODBC pour la sortie spécifiant que la connexion doit être ouverte en tant que connexion Unicode.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

Format : ODBCOPENEx (nom de l'ensemble de données, nom du client de l'ensemble de données, mot de passe du client, (indicateur use-Unicode-interface) )

```
ODBCOpenEx(Source, ClientName, Password, UseUnicodeODBC);
```

Argument	Description
Source	Nom d'une source de données ODBC.
ClientName	Client valide sur la source de données.
Password	Mot de passe pour ClientName.
UseUnicodeODBC	Définit le type de connexion Unicode à utiliser.

### Exemple

```
ODBCOpenEx( TestTable, sa, , 1 );
```

```
chinese= ;
chinese = CHARW( 37123 );
fieldval = chinese | SomeNewText;
sql = Update TestTable set ForeName = N | fieldval | WHERE CustomerId = 1
ODBCOUTPUT( Unicode, sql );
```

L'instruction SQL obtenue a l'apparence suivante :

```
Update TestTable set ForeName = N?SomeNewText WHERE CustomerId = 1
```

### ODBCOutput

ODBCOutput exécute une requête de mise à jour SQL sur une source de données ODBC ouverte. Il est recommandé d'utiliser la fonction ODBCOpen pour ouvrir la source de données avant d'appeler ODBCOutput et d'utiliser ODBCclose pour fermer la source de données avant de quitter le processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ODBCOutput(Source, SQLQuery, [SQLQuery2, SQLQuery3, ...]);
```

Argument	Description
Source	Source de données ODBC où vous voulez exécuter une requête.
SQLQuery	<p>Instruction de requête SQL.</p> <p>Bien qu'ODBCOutput ait été créée pour servir à la mise à jour des tables, elle peut être utilisée pour exécuter toute requête SQL sur la source de données.</p> <p>Dans les cas où l'instruction de requête SQL dépasse 255 caractères, vous devez la diviser en plusieurs arguments SQLQuery (SQLQuery2, SQLQuery3, etc.). Vous pouvez ainsi créer des instructions de requête dépassant la limite de 255 caractères pour les arguments TurboIntegrator. Quand la fonction ODBCOutput est exécutée, tous les arguments SQLQuery sont concaténés et la requête est exécutée.</p>

## Exemple

```
ODBCOutput('Comptabilité', 'INSERT [CodeCatégorie], [NomCatégorie]Catégories FROM;');
```

Cette fonction exécute une requête SQL spécifiée sur la source de données Accounting.

## SetODBCUnicodeInterface

SetODBCUnicodeInterface indique si l'interface ODBC doit utiliser les fonctions Unicode "larges" ou les fonctions standard à caractères mono-octet. Lorsque cette fonction a pour valeur 1, l'interface ODBC à caractères larges est utilisée.

Certains pilotes ODBC prennent en charge à la fois les anciennes interfaces à caractères codés sur un seul octet, et les interfaces à caractères larges de style Unicode où les caractères sont passés et extraits en tant que quantités à 16 bits. Si le pilote choisi ne prend pas l'un ou l'autre de ces types d'interface, un indicateur impose à TurboIntegrator l'utilisation d'un style d'interface donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SetODBCUnicodeInterface=1
```

Argument	Description
1	Utilise l'interface ODBC à caractères larges.
0	Utilise l'interface à caractères codés sur un octet.

## Fonctions TurboIntegrator de contrôle de processus

Ces fonctions sont utilisées dans le contrôle des processus.

### ExecuteCommand

ExecuteCommand exécute une ligne de commande pendant un processus. Vous pouvez utiliser la fonction ExecuteCommand pour exécuter une application de bureau, mais pas un service.

Si vous utilisez la fonction ExecuteCommand pour exécuter un fichier exécutable, les conditions suivantes s'appliquent :

- Si l'argument CommandLine spécifie uniquement le nom du fichier à exécuter, un serveur Windows recherche le fichier à la fois dans le répertoire de base de données du serveur et dans le répertoire dans lequel réside Tm1s.exe. Un serveur UNIX recherche le fichier uniquement dans le répertoire de base de données du serveur.
- Si l'argument CommandLine utilise un préfixe de chemin relatif, le serveur Windows et le serveur UNIX tentent tous deux de localiser le fichier dans le seul répertoire de données du serveur.
- Vous pouvez passer un chemin absolu à l'argument CommandLine sur le serveur Microsoft Windows ou UNIX pour exécuter un fichier à n'importe quel emplacement.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ExecuteCommand(CommandLine, Wait);
```



Argument	Description
CommandLine	Ligne de commande à exécuter.
Wait	Indique si le processus doit attendre la fin de l'exécution avant de passer à l'instruction de processus suivante. La valeur d'argument 0 commande au processus de passer à l'instruction suivante sans attendre l'exécution de la ligne de commande. La valeur d'argument 1 commande au processus d'attendre la fin de l'exécution de la ligne de commande avant de passer à l'instruction suivante.

## ExecuteProcess

ExecuteProcess permet d'exécuter un processus TurboIntegrator à partir d'un autre processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ExecuteProcess(ProcessName, [ParamName1, ParamValue1, ParamName2, ParamValue2]);
```

Argument	Description
ProcessName	Nom du processus à exécuter. Ce processus doit résider sur le même serveur que le processus à partir duquel ExecuteProcess est appelé.  Si le processus nommé par cet argument n'est pas trouvé lors de l'exécution, le processus d'appel est immédiatement arrêté. (TurboIntegrator ne vérifie pas s'il existe un nom de processus ProcessName valide lors de la compilation.)
ParamName	Nom d'un paramètre existant du processus à exécuter. Cet argument est requis uniquement si le processus à exécuter utilise des paramètres.
ParamValue	Valeur valide pour le paramètre ParamName. Si vous spécifiez un argument ParamName, vous devez spécifier une ParamValue correspondante.  Les arguments ParamName et ParamValue doivent apparaître en paires ordonnées, avec le nom du paramètre suivi de la valeur. Vous devez spécifier un ParamName et une ParamValue correspondante pour chaque paramètre du processus à exécuter.

Les noms de paramètre passés dans la fonction ExecuteProcess sont comparés lors de l'exécution avec les noms de paramètre spécifiés dans le processus à exécuter. Si les noms passés ne sont pas trouvés dans la liste de paramètres du processus à exécuter, une erreur grave se produit, entraînant l'arrêt immédiat du processus à partir duquel ExecuteProcess est appelé.

### Valeurs renvoyées

La fonction ExecuteProcess renvoie une valeur réelle qui peut être testée avec une des fonctions de valeur renvoyée suivantes :

Fonction	Description
ProcessExitByChoreQuit()	Indique que le processus s'est fermé en raison de l'exécution de la fonction ChoreQuit.
ProcessExitNormal()	Indique que le processus a été exécuté normalement.
ProcessExitMinorError()	Indique que le processus a été exécuté avec succès mais a rencontré des erreurs mineures.
ProcessExitByQuit()	Indique la sortie du processus en raison d'une commande "quit" explicite.
ProcessExitWithMessage()	Indique la sortie normale du processus, avec un message écrit dans <code>tm1server.log</code>
ProcessExitSeriousError()	Indique la sortie du processus en raison d'une erreur grave.
ProcessExitOnInit()	Indique que le processus a été abandonné pendant l'initialisation.
ProcessExitByBreak()	indique la sortie du processus sur détection de fonction ProcessBreak.

### Exemple

Si vous voulez enregistrer quand un processus appelé par `ExecuteProcess` échoue à cause d'une erreur grave, utilisez un code similaire au suivant :

```
return_value = ExecuteProcess('create_sales_cube');
ASCIIOutput('C:\temp\process_return_value.txt', 'Le processus s'est arrêté
avec des erreurs sérieuses à ', TIME, 'le', TODAY);if(return_value = ProcessExitSeriousError() )
endif;
```

### GetProcessErrorFileDirectory

`GetProcessErrorFileDirectory` renvoie le chemin complet, avec une barre oblique de fin, du répertoire dans lequel les fichiers d'erreurs de processus TurboIntegrator sont générés. Par défaut, tous les fichiers journaux d'erreurs de processus sont écrits dans le répertoire de données du serveur sur lequel le processus réside.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
GetProcessErrorFileDirectory;
```

### Arguments

Aucun.

### GetProcessErrorFilename

`GetProcessErrorFilename` renvoie le nom du fichier journal d'erreurs d'un processus TurboIntegrator associé à un processus. Si le processus n'a pas encore généré de fichier journal d'erreurs, la fonction renvoie une chaîne vide (nulle).

**Important :** Un fichier journal d'erreurs de processus n'est pas généré tant que toutes les instructions d'un onglet de processus donné (Prologue, Métadonnées, Données ou Epilogue) n'ont pas été exécutées. Vous pouvez donc utiliser `GetProcessErrorFilename` pour vérifier si des onglets précédents ont généré un

fichier journal d'erreurs mais vous ne pouvez pas utiliser cette fonction pour déterminer si l'onglet de processus en cours provoque l'écriture d'erreurs dans un fichier journal.

Par exemple, en déterminant que `GetProcessErrorFilename` renvoie une chaîne non nulle dans l'onglet Epilogue, vous savez que des erreurs ont été générées dans les onglets Prologue, Métadonnées ou Données. Cependant, vous ne pouvez pas utiliser `GetProcessErrorFilename` dans l'onglet Data pour déterminer si ce dernier génère des erreurs.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
GetProcessErrorFilename;
```

### Arguments

Aucun.

## GetProcessName

`GetProcessName` renvoie le nom du processus en cours sous forme de chaîne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
GetProcessName()
```

### Arguments

Aucun.

```
Name = GetProcessName();
```

## If

L'instruction `If` permet à un processus d'exécuter une instruction ou une série d'instructions quand une expression donnée est vraie. Vous pouvez utiliser des opérateurs arithmétiques, opérateurs logiques et opérateurs de comparaison pour construire une expression.

L'instruction `If` de TurboIntegrator diffère de la fonction de règles IF en ce que l'instruction TurboIntegrator peut accepter plusieurs instructions `ElseIf` ou `Else` pour évaluer des expressions multiples, tandis que la fonction de règles IF ne peut évaluer qu'une seule expression.

Vous pouvez emboîter jusqu'à 20 instructions `If/ElseIf/Else` dans un processus TurboIntegrator. Si vous dépassez 20 instructions `If/ElseIf/Else` emboîtées, vous recevez une erreur en tentant d'enregistrer le processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
If(expression);  
statement1;  
ElseIf(expression);  
statement2;  
ElseIf(expression);  
statement3;  
Else;  
statement4;  
EndIf;
```

## Arguments

Aucun.

## Exemples

```
If (x=5);
    ASCIIOutput('c:\temp\if.txt','x égale cinq');
ElseIf (x=1);
    ASCIIOutput ('c:\temp\if.txt', 'x égale un');
ElseIf (x=2);
    ASCIIOutput ('c:\temp\if.txt', 'x égale deux ');
ElseIf (x=3);
    ASCIIOutput ('c:\temp\if.txt', 'x égale trois');
ElseIf (x=4);
    ASCIIOutput ('c:\temp\if.txt', 'x égale quatre');
Else;
    ASCIIOutput ('c:\temp\if.txt', 'x falls outside expected range');
EndIf;
```

Cet exemple évalue la valeur de X. Si X=5, la fonction ASCIIOutput est exécutée pour écrire la chaîne "x égale cinq" dans c:\temp\if.txt. Si X n'est pas égal à 5, la première instruction ElseIf est évaluée. Si X=1, la fonction ASCIIOutput est exécutée pour écrire la chaîne "x égale un" dans c:\temp\if.txt. Ce traitement se poursuit jusqu'à l'exécution de EndIf.

Les instructions If peuvent également être construites sans ElseIf, comme dans l'exemple suivant :

```
IF(expression);
    statement1;
ELSE;
    statement2;
ENDIF;
```

## ItemReject

ItemReject rejette un enregistrement source et le place dans le journal des erreurs avec un message d'erreur spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ItemReject(ErrorString);
```

Argument	Description
ErrorString	Message d'erreur à consigner dans le journal des erreurs lorsqu'un enregistrement est rejeté.

### Exemple

```
ItemReject(' Valeur hors de la plage acceptable.');
```

Cet exemple place un enregistrement source dans le journal des erreurs accompagné du message d'erreur 'Valeur hors de la plage acceptable' lorsque l'enregistrement source contient une valeur non comprise dans la plage définie.

## ItemSkip

ItemSkip force un processus à ignorer l'élément de source de données en cours.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ItemSkip;
```

## Arguments

Aucun.

## ProcessBreak

ProcessBreak arrête le traitement des données de la source et passe à la partie Epilogue d'un processus. Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ProcessBreak;
```

## Arguments

Aucun.

## ProcessError

ProcessError entraîne l'arrêt immédiat d'un processus. Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ProcessError;
```

## Arguments

Aucun.

## ProcessExists

ProcessExists détermine si un processus TurboIntegrator spécifique existe. La fonction ProcessExists peut renvoyer l'une des trois valeurs suivantes :

- Si un processus TurboIntegrator du nom spécifié n'existe pas, la fonction renvoie 0.
- Si un processus du nom spécifié existe et est valide, la fonction renvoie 1.
- Si un processus du nom spécifié existe mais présente des erreurs de compilation, la fonction renvoie -1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ProcessExists(ProcessName);
```

Argument	Description
ProcessName	Nom du processus dont vous essayez de déterminer le statut.

## ProcessQuit

ProcessQuit arrête un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessQuit;
```

### Arguments

Aucun.

## ProcessRollback

ProcessRollback lance l'annulation d'un processus. Si elle est utilisée dans un processus TurboIntegrator, cette fonction rejette toutes les éditions en attente et annule les traitements supplémentaires. Un message d'erreur apparaît dans les fichiers tm1server.log et tm1processorerrorXXX.log.

**Remarque :** Dans IBM Planning Analytics version 2.0.8 ou ultérieure, lorsqu'un processus TurboIntegrator est annulé et redémarré, le processus est désormais représenté dans le fichier tm1server.log sous la forme de trois étapes : démarrage, redémarrage en raison de conflits d'accès ou d'une annulation, puis arrêt.

Une entrée est ajoutée au fichier tm1server.log, qui indique que le processus TurboIntegrator est en cours de redémarrage en raison d'un conflit d'accès ou d'une annulation au lieu d'un simple démarrage. Cette consignation est activée par défaut sans définir d'options de débogage spécifiques.

Si elle est utilisée dans une tâche en mode validation unique, la fonction ProcessRollback rejette toutes les modifications en attente de tous les processus précédents et poursuit son exécution à partir du prochain processus de la tâche. En cas de conflit d'accès après l'appel de ProcessRollback, toute la tâche est redémarrée.

Si elle est utilisée dans une tâche en mode validations multiples, ProcessRollback rejette toutes les modifications en attente du processus en cours, puis poursuit son exécution à partir du prochain processus de la tâche. Les modifications déjà validées ne peuvent pas être annulées. En cas de conflit d'accès après l'appel de ProcessRollback, seul le processus en cours est redémarré.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ProcessRollback;
```

### Arguments

Aucun.

## RunProcess

RunProcess permet d'exécuter des processus TurboIntegrator en parallèle, chacun dans sa propre unité d'exécution gérée par TM1 Server. Cette approche accélère le chargement des données et d'autres opérations dans lesquelles des processus TurboIntegrator sont utilisés pour diviser le travail.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
RunProcess(ProcessName, [ParamName1, ParamValue1, ParamName2, ParamValue2]);
```

Argument	Description
ProcessName	Nom du processus à exécuter. Ce processus doit se trouver sur le même serveur que le processus à partir duquel RunProcess est appelé.  Si le processus nommé par cet argument n'est pas trouvé lors de l'exécution, le processus d'appel est immédiatement arrêté. (TurboIntegrator ne vérifie pas s'il existe un nom de processus ProcessName valide lors de la compilation.)
ParamName	Nom d'un paramètre existant du processus à exécuter. Cet argument est obligatoire uniquement si le processus à exécuter utilise des paramètres.
ParamValue	Valeur valide pour le paramètre ParamName. Si vous spécifiez un argument ParamName, vous devez spécifier une ParamValue correspondante.  Les arguments ParamName et ParamValue doivent apparaître en paires ordonnées, avec le nom du paramètre suivi de la valeur. Vous devez spécifier une valeur ParamName et la valeur ParamValue correspondante pour chaque paramètre du processus à exécuter.

Les noms de paramètre transmis dans la fonction RunProcess sont comparés lors de l'exécution aux noms de paramètre spécifiés dans le processus à exécuter. Si les noms transmis ne figurent pas dans la liste de paramètres du processus à exécuter, une erreur grave se produit et entraîne l'arrêt immédiat du processus à partir duquel RunProcess est appelé.

### Valeurs renvoyées

RunProcess renvoie une chaîne. La chaîne correspond à l'ID de travail ou à une chaîne vide si une erreur se produit.

## Synchronized

Synchronized est utilisé dans un script TurboIntegrator pour forcer l'exécution en série d'un ensemble désigné de processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

La syntaxe de la fonction synchronized() est la suivante :

```
synchronized (lockName);
```

Synchronized n'a qu'un seul paramètre obligatoire, qui est le nom défini par l'utilisateur d'un objet de verrouillage. Le nom de cet objet de verrouillage peut être utilisé dans plusieurs processus TurboIntegrator pour sérialiser leur exécution en tant que groupe.

Argument	Description
lockName	Nom, défini par l'utilisateur, d'un objet de verrouillage sur lequel baser la synchronisation. Les noms sont insensibles à la casse et les blancs sont ignorés. Leur longueur totale ne doit pas dépasser 1023 caractères.  Chaîne/Oui/Aucune

## Sémantique

Un processus TurboIntegrator peut contenir un nombre quelconque d'appels à `synchronized()`, avec un nombre quelconque d'objets de verrouillage. La sérialisation est effective depuis le moment où la fonction `synchronized()` est appelée, jusqu'à la fin de la transaction qui la contient.

Par exemple, si `synchronized()` est appelée depuis le sous-processus (Ps) d'un processus maître (Pm) ou d'une tâche maîtresse (Tm), l'objet de verrouillage est "libéré" lorsque Pm ou Tm prend fin. Il existe une exception : `SaveDataAll (SDA)` "termine" prématurément l'exécution de milieu de processus d'une transaction. Ceci s'applique également aux objets de verrouillage.

L'appel à `synchronized()` peut être placé à n'importe quel endroit dans un script TurboIntegrator, mais, lorsqu'il est détecté, la sérialisation s'applique à l'ensemble du processus TurboIntegrator.

Prenons le cas d'un processus TurboIntegrator dont le script contient, à peu près au centre, un appel à `synchronized()` précédé de l'opération O1. Deux instances de ce processus TurboIntegrator peuvent démarrer en même temps. Il est possible qu'une instance s'achève, et achève son appel à `synchronized()`, avant que la seconde instance n'atteigne le sien. Dans ce cas, l'utilisateur voit les deux processus comme simultanés. Si, à la place, le second processus atteint son appel à `synchronized()` avant la fin du premier processus, il défait tout ce qui avait été fait (O1) et attend la fin du premier processus. Dans ce cas, l'utilisateur voit les deux processus comme sérialisés.

Pour éviter cette confusion et optimiser l'utilisation de `synchronized()`, il est recommandé (mais pas obligatoire) d'entrer les appels à `synchronized()` comme premières instructions d'un processus TurboIntegrator.

## Exemple

Supposons que le processus TurboIntegrator P ait besoin de mettre à jour deux cubes, `Cube_1` et `Cube_2`.

D'autres processus TurboIntegrator peuvent aussi avoir besoin de modifier `Cube_1` ou `Cube_2`.

Pour obliger tous les processus TurboIntegrator qui doivent mettre à jour `Cube_1` ou `Cube_2` à s'exécuter un par un, P pourrait appeler `synchronized()` de la façon suivante :

```
sCube_1='Cube_1';
sCube_2='Cube_2';
sE1='Elm1';
sE2='Elm2';
sE4='Units';
sE5='Price';

Synchronized( sCube_1 );
Synchronized( sCube_2 );

CellPutn( 111, sCube_1, sE1, sE2 );
CellPutn( 9.99, sCube_2, sE4, sE5 );

# ...
```

Les autres processus TurboIntegrator qui mettent à jour `Cube_1` ou `Cube_2` doivent aussi appeler `synchronized( sCube_1 )` et/ou `synchronized( sCube_2 )` de la même manière.

Dans cet exemple, les noms des deux objets de verrouillage ont été choisis pour être identiques aux noms des cubes. Mais le nom des objets de verrouillage ne doit pas obligatoirement être le même que ceux des autres objets TM1 (cubes, dimensions, sous-ensembles).

## While

L'instruction `While` permet à un processus de répéter une série d'instructions tant qu'une condition donnée est vraie. Les instructions `While` peuvent être imbriquées.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



## Syntaxe

```
WHILE(logical expression);  
statement1;  
statement2;  
...  
statement n;  
END;
```

**Remarque :** Notez que toutes les instructions WHILE doivent se terminer par une instruction END.

## Arguments

Aucun.

## Fonctions TurboIntegrator de gestion de règles

---

Ces fonctions facilitent la gestion des règles.

### CubeProcessFeeders

CubeProcessFeeders traite une nouvelle fois tous les feeders des règles pour un cube spécifié.

Cette fonction traite de nouveau tous les feeders des règles pour un cube spécifié. En principe, les feeders font automatiquement l'objet d'un nouveau traitement lors de l'enregistrement des modifications apportées à un fichier de règles. Toutefois, si les données changent et que ces modifications influent sur les feeders conditionnels, cette fonction sera appelée pour réévaluer les feeders conditionnels.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeProcessFeeders(CubeName);
```

Argument	Description
CubeName	Cube pour lequel vous voulez retraiter des feeders.

## Exemple

```
CubeProcessFeeders('ventes2003');
```

Cette fonction traite de nouveau tous les feeders des règles pour le cube 2003sales.

### CubeRuleAppend

CubeRuleAppend ajoute une seule ligne de texte à une règle de cube TM1.

Principalement, cette fonction ajoute une seule ligne de texte à un fichier de règle (.rux). La ligne de texte peut être une instruction, mais aussi un commentaire. Si aucune règle n'est associée au cube lorsque la fonction est exécutée, une règle contenant la ligne transmise est créée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeRuleAppend(CubeName, RuleText, IsCalculationRule);
```

Argument	Description
CubeName	Nom du cube associé à la règle à laquelle vous voulez ajouter une ligne de texte.
RuleText	<p>Ligne de texte à ajouter à la règle.</p> <p>L'ensemble de la ligne de texte à ajouter doit être encadré par des guillemets simples, et doit respecter les conventions de syntaxe des règles.</p> <p>Si la ligne contient des références d'élément, les noms d'élément doivent être encadrés par deux guillemets à gauche et à droite pour échapper les guillemets simples qui encadrent normalement les noms d'élément. Par exemple, la référence à un élément nommé CL3 doit être entrée sous la forme [' 'CL3 ' '].</p> <p>Les lignes qui suivent sont des exemples de lignes de texte valides que vous pouvez ajouter à une règle :</p> <pre>' [' 'CL3 ' ' ] = [ 'CL4 ' ] + [ 'Trial ' ];' ' skipcheck; ' ' [' 'Trial ' ' ] =&gt; [ 'CL3 ' ' ];'</pre>
IsCalculationRule	<p>Le paramètre IsCalculationRule déclare si la ligne doit être insérée juste au-dessus de toute section feeder existante dans la règle de cube. Si le paramètre IsCalculationRule est omis, ou si sa valeur transmise est 0.0, la nouvelle ligne est simplement ajoutée à la fin de la règle.</p> <p>Les fichiers de règle (.rux) étant constitués par une section de calcul suivie d'une section feeder facultative, la valeur 1.0 doit être utilisée pour l'ajout des lignes d'instruction de règle de calcul, afin qu'elle soient ajoutées au bon endroit du fichier de règle.</p>

## Exemples

```
CubeRuleAppend( 'MyCube', [ 'CL3 ' ] = [ 'CL4 ' ] + [ 'Trial ' ];', 1.0 );
```

Dans cet exemple, l'instruction de calcul [ 'CL3 ' ] = [ 'CL4 ' ] + [ 'Trial ' ]; est ajoutée à la fin de la section de calcul de la règle du cube MyCube.

```
CubeRuleAppend( 'MyCube', [ 'Trial ' ] => [ 'CL3 ' ];', 0.0 );
```

Dans cet exemple, l'instruction FEEDERS [ 'Trial ' ] => [ 'CL3 ' ]; est ajoutée à la fin de la règle du cube MyCube.

## CubeRuleDestroy

CubeRuleDestroy supprime toute règle existante pour un cube spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
CubeRuleDestroy(CubeName);
```

Argument	Description
CubeName	Nom du cube associé à la règle à supprimer.

## Exemple

```
CubeRuleDestroy('SalesProjections');
```

Dans cet exemple, la règle du cube SalesProjections est supprimée.

## DeleteAllPersistentFeeders

DeleteAllPersistentFeeders supprime tous les fichiers .feeder qui ont été conservés. Lorsque cette fonction est utilisée, tous les cubes sont associés à la mention "do not save feeders" afin qu'un processus SaveData ne conserve pas les feeders et que tous les feeders soient recalculés lors du redémarrage du serveur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
DeleteAllPersistentFeeders;
```

## Arguments

Aucun.

## ForceSkipCheck

ForceSkipCheck force la requête à se comporter comme si le cube contenait une instruction SKIPCHECK dans ses règles.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Ceci signifie que la requête ne traitera que les valeurs effectivement dans le cube (par opposition à une situation sans SKIPCHECK où chaque cellule serait énumérée en vue de rechercher des valeurs). Cette fonction définit l'état de la requête de vue de sorte à sélectionner uniquement les valeurs dans le cube. Cette fonction doit être ajoutée dans la section Prologue du processus TurboIntegrator. Le positionnement de ForceSkipCheck() dans le prologue affecte la requête de vue complète des éléments de données qui suivent.

## Syntaxe

```
ForceSkipCheck()
```

## Arguments

Aucun.

## RuleLoadFromFile

RuleLoadFromFile crée une règle TM1 pour un cube spécifié depuis un fichier texte. Chaque instruction de règle doit se terminer par un point-virgule (;) et les commentaires doivent être précédés du caractère #. Si une règle existe déjà pour le cube spécifié, elle est remplacée par la règle créée par RuleLoadFromFile.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
RuleLoadFromFile(Cube, TextFile);
```

Argument	Description
Cube	Nom du cube pour lequel vous voulez créer une règle.
TextFile	<p>Nom du fichier texte à partir duquel vous voulez créer une règle.</p> <p>Vous pouvez spécifier le chemin complet de ce fichier, avec le nom du fichier et l'extension. (Exemple 1 ci-dessous.)</p> <p>Si vous spécifiez uniquement le nom et l'extension du fichier, TurboIntegrator recherche ce fichier dans le répertoire de données du serveur.</p> <p>Si vous ne spécifiez pas d'extension de fichier, TurboIntegrator suppose l'extension .rux par défaut. (Exemple 2 ci-dessous.)</p>

Si vous laissez vide l'argument TextFile, TurboIntegrator recherche un fichier source portant le même nom que le cube (mais avec l'extension .rux) dans le répertoire de données du serveur. (Exemple 3 ci-dessous.)

### Exemple

```
RuleLoadFromFile('Sales', 'C:\temp\cuberule.txt');
```

Cet exemple utilise le contenu du fichier cuberule.txt du répertoire C:\temp pour créer une règle pour le cube Sales.

```
RuleLoadFromFile('Sales', 'cuberule');
```

Cet exemple crée une règle pour le cube Sales à l'aide du fichier cuberule.rux du répertoire de données du serveur.

```
RuleLoadFromFile('Sales', '');
```

Cet exemple crée une règle pour le cube Sales à l'aide du fichier Sales.rux du répertoire de données du serveur.

## Fonctions de bac à sable

Ces fonctions sont utilisées avec les bacs à sable.

### GetUseActiveSandboxProperty

GetUseActiveSandboxProperty renvoie une valeur booléenne qui indique si un processus lit et écrit les données dans les données de base ou dans le bac à sable actif de l'utilisateur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Par défaut, les processus lisent et écrivent dans les données de base.

- Si la valeur renvoyée est 0, le processus lit et écrit les données dans les données de base.
- Si la valeur renvoyée est 1, le processus lit et écrit les données dans le bac à sable actif.

**Remarque :** Cette fonction renvoie la valeur permanente de cette propriété, telle que définie dans l'interface utilisateur de l'Explorateur de serveur / Architect, *sauf* si vous avez utilisé la fonction `SetActiveSandboxProperty` dans le processus. Dans ce cas, la valeur de cette propriété est déterminée par la valeur qui a été définie au moyen de la fonction `SetActiveSandboxProperty`.

### Syntaxe

```
GetUseActiveSandboxProperty()
```

### Arguments

Aucun.

### Exemple

```
return_value = GetUseActiveSandboxProperty();
```

Cet exemple renvoie une valeur booléenne indiquant si le processus lit et écrit actuellement les données de cube dans le bac à sable actif ou dans les données de base.

## ServerActiveSandboxGet

`ServerActiveSandboxGet` renvoie le nom du bac à sable actif de l'utilisateur. Si l'utilisateur n'a pas de bac à sable actif, une chaîne vide est renvoyée. Dans la mesure où les tâches sont exécutées dans le contexte d'un utilisateur administrateur spécial et qu'elles peuvent n'avoir aucun bac à sable actif, cette fonction renvoie toujours une chaîne vide lorsqu'elle est exécutée à l'aide d'une tâche.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerActiveSandboxGet();
```

### Arguments

Aucun.

### Exemple

```
return_value = ServerActiveSandboxGet();
```

Cet exemple renvoie le bac à sable actif de l'utilisateur qui exécute le processus TurboIntegrator dans lequel l'appel de fonction est effectué.

## ServerActiveSandboxSet

`ServerActiveSandboxSet` définit le bac à sable actif de l'utilisateur qui l'exécute. Une chaîne vide sert à effacer le bac à sable actif de l'utilisateur. Cette fonction renvoie une erreur si l'utilisateur qui l'exécute ne possède pas de bac à sable portant le nom indiqué.

Dans la mesure où les tâches sont exécutées dans le contexte d'un utilisateur administrateur spécial et qu'elles peuvent n'avoir aucun bac à sable actif, cette fonction renvoie toujours une erreur lorsqu'elle est exécutée à l'aide d'une tâche.

**Remarque :** Pour qu'un processus TurboIntegrator puisse lire et écrire des valeurs dans le contexte du bac à sable actif de l'utilisateur exécutant la fonction, la propriété `UseActiveSandbox` doit être activée. Voir «[GetUseActiveSandboxProperty](#)», à la page 348 et «[SetActiveSandboxProperty](#)», à la page 358.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ServerActiveSandboxSet(SandboxName)
```

Argument	Description
SandboxName	Valeur de chaîne. Nom d'un bac à sable appartenant à l'utilisateur qui exécute la fonction.

**Exemple : Affectation de la valeur "Best case" au bac à sable actif de l'utilisateur qui exécute l'action.**

```
ServerActiveSandboxSet('Best case');
```

**Exemple : Effacement du bac à sable actif de l'utilisateur qui exécute l'action et restauration du contexte sous la forme de données de base.**

```
ServerActiveSandboxSet('');
```

## ServerSandboxClone

ServerSandboxClone clone un bac à sable existant pour obtenir un nouveau bac à sable.

Les bacs à sable sont des espaces de travail privés dans lesquels un utilisateur peut saisir et stocker des valeurs de données, séparément des données de base de TM1. Les bacs à sable sont stockés sur le disque et, lorsqu'ils sont utilisés, en mémoire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxClone(nom_bac_à_sable,nom_bac_à_sable );
```

Argument	Description
<i>nom_bac_à_sable</i>	Valeur de chaîne. Nom d'un bac à sable appartenant à l'utilisateur qui exécute la fonction.
<i>nom_bac_à_sable</i>	Valeur de chaîne. Nom d'un bac à sable à créer en tant que clone de <i>nom_bac_à_sable</i> .

### Exemple

```
ServerSandboxClone( 'Best case', 'Second best case');
```

## ServerSandboxCreate

ServerSandboxCreate crée un bac à sable.

Les bacs à sable sont des espaces de travail privés dans lesquels un utilisateur peut saisir et stocker des valeurs de données, séparément des données de base de TM1. Les bacs à sable sont stockés sur le disque et, lorsqu'ils sont utilisés, en mémoire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxCreate( nom_bac_à_sable );
```

Argument	Description
nom_bac_à_sable	Valeur de chaîne. Nom d'un bac à sable à créer.

### Exemple

```
ServerSandboxCreate( 'My sandbox' );
```

## ServerSandboxesDelete

La fonctionnalité ServerSandboxesDelete permet aux administrateurs de supprimer des bacs à sable d'utilisateurs correspondant à certains critères.

Les bacs à sable sont des espaces de travail privés dans lesquels un utilisateur peut saisir et stocker des valeurs de données, séparément des données de base de TM1. Les bacs à sable sont stockés sur le disque et, lorsqu'ils sont utilisés, en mémoire.

Cette fonction est exécutée côté serveur et est disponible via TurboIntegrator et la fonction d'API ServerSandboxesDelete. Si elle est utilisée dans un processus TurboIntegrator, les administrateurs peuvent planifier la maintenance au moyen de tâches automatisées.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Description

Cette fonctionnalité utilise un "prédicat" pour décrire le bac à sable en cours de suppression. Un prédicat peut être lu de la façon suivante : "Supprimer les bacs à sable dont l'*attribut* est *condition valeur*".

Par exemple : "Supprimer les bacs à sable dont la taille est supérieure à 10 Mo". Dans cet exemple, l'attribut est la "taille (size)" du bac à sable, la condition est "supérieur(e) à" et la valeur est "10 Mo".

La fonction TurboIntegrator comporte deux paramètres de délimiteur facultatifs. Etant donné que le type de délimiteur utilisé dans le nom d'un bac à sable n'est soumis à aucune restriction, les administrateurs peuvent choisir leur propre délimiteur "sécurisé".

Par exemple, `ServerSandboxesDelete( 'client:=Admin, name:=:best case scenario' );`

Dans l'exemple suivant, le signe deux-points est utilisé dans le nom du bac à sable ("best::case::scenario"), ce qui nécessite un autre délimiteur :

```
ServerSandboxesDelete( 'client|=|Admin# name|=|best::case::scenario', '|', '#' );
```

**Remarque :** La syntaxe exacte d'un prédicat est différente entre la forme TurboIntegrator et API de cette fonction.

### Syntaxe

```
ServerSandboxesDelete(string,string,string)
```

Argument	Description
Prédicats	Nom du processus à exécuter. Ce processus doit se trouver sur le même serveur que le processus à partir duquel RunProcess est appelé.  Requis String Aucune valeur par défaut  Liste des prédicats de longueur arbitraire. Chaque prédicat est une chaîne contenant trois jetons. Le premier jeton indique un attribut d'un bac à sable. Le second indique une condition, par exemple ">" ou "=". Le troisième jeton est une valeur possible de l'attribut sur lequel les bacs à sable doivent être filtrés de façon conditionnelle. La longueur de la chaîne complète ne doit pas dépasser 10 000 caractères.
PredicateDelimiter	Facultatif String La valeur par défaut est le signe deux-points (:) Délimiteur facultatif. La longueur de la chaîne ne doit pas dépasser 1 caractère.
PredicateListDelimiter	Facultatif String La valeur par défaut est la virgule (,) Délimiteur facultatif. La longueur de la chaîne ne doit pas dépasser 1 caractère.

### Attributs de filtre

Les attributs de filtre sont les propriétés d'un bac à sable sur lesquelles une correspondance conditionnelle peut être établie. Les noms d'attribut et leurs conditions valides correspondantes sont insensibles à la casse et ignorent les blancs incorporés. Par exemple, les deux appels suivants sont tous les deux valides :

```
ServerSandboxesDelete( 'client::Admin' );
```

```
ServerSandboxesDelete( 'C L I E N T : = :Admin' );
```

Tableau 1. Attributs de filtre			
Attribut	Description	Conditions valides	Type de valeur
UpdateDate	Horodatage de la dernière écriture effectuée dans le bac à sable.	<, =, >.	Horodatage exprimé dans le format international standard (par exemple, AAAA-MM-JJ). Les jours sont les unités les plus précises.
AccessDate	Horodatage du dernier déchargement d'un bac à sable.	<, =, >.	Horodatage exprimé dans le format international standard (par exemple, AAAA-MM-JJ). Les jours sont les unités les plus précises.



Tableau 1. Attributs de filtre (suite)

Attribut	Description	Conditions valides	Type de valeur
CreationDate	Horodatage de la création d'un bac à sable.	<, =, >.	Horodatage exprimé dans le format international standard (par exemple, AAAA-MM-JJ). Les jours sont les unités les plus précises.
Size	Taille en mémoire d'un bac à sable.	<, =, >.	Taille respectant les règles de conversion de log4cxx (voir le paramètre de configuration AuditLogMaxTemp FileSize). Par exemple, 10 Mo. Les kilooctets sont les unités les plus précises.
Name	Nom d'un bac à sable.	=, containing.	Chaîne.
Client	Client propriétaire d'un bac à sable.	=.	Chaîne.
Group	Groupe auquel appartient le client propriétaire d'un bac à sable.	=.	Chaîne.

### Journalisation et retours

La suppression des bacs à sable est enregistrée à l'aide de la fonctionnalité de journalisation d'audit préexistante. En outre, un état plus détaillé des effets de l'administration des bacs à sable est inclus dans le fichier journal de débogage (tm1server.log) au niveau INFO. Cet état contient la liste des bacs à sable affectés, certains de leurs attributs ainsi que les éventuelles erreurs rencontrées.

ServerSandboxesDelete renvoie uniquement l'état de succès ou d'échec.

### Sémantique

#### Liste des prédicats

Plusieurs prédicats transmis dans un même appel à ServerSandboxesDelete sont conjonctifs. Autrement dit, pour qu'un bac à sable corresponde aux critères indiqués, tous les prédicats doivent être vrais. Il est possible d'utiliser plusieurs appels à ServerSandboxesDelete pour obtenir le comportement disjonctif. Une seule occurrence de chaque attribut est autorisée par appel à ServerSandboxesDelete. Par exemple, il est incorrect de passer un client deux fois car un bac à sable ne possède qu'un seul client propriétaire. Lorsque plusieurs occurrences d'un attribut sont détectées, un avertissement s'affiche dans l'état détaillé sans toutefois que l'opération ne soit abandonnée sur un échec. Dans ce cas, les prédicats sont testés comme avec n'importe quelle autre requête mais l'ensemble de résultats est toujours vide.

#### Verrouillage

Pour éviter les problèmes de verrouillage massifs, ServerSandboxesDelete examine les bacs à sable d'un client comme une image instantanée de point de cohérence puis, lorsque c'est possible, libère tous les verrous qui garantiraient une transaction sérialisable. En raison de ce comportement, une fois qu'un client est "passé" dans l'itération de tous les clients, il est possible de lui ajouter un bac à sable correspondant aux critères de filtre avant la fin de la transaction de maintenance. C'est comme si vous ajoutiez un bac à sable au client immédiatement après l'achèvement de la transaction.

#### Portée

Les membres des groupes ADMIN (super utilisateur) et DataAdmin peuvent accéder à tous les bacs à sable de tous les clients. Ils doivent spécifier explicitement l'attribut du client afin de limiter la portée de leur appel à ServerSandboxesDelete à leurs propres bacs à sable. Tous les autres utilisateurs ne

peuvent accéder qu'à leurs propres bacs à sable. S'ils spécifient un autre client ou un groupe auquel ils n'appartiennent pas, la fonction s'arrête sur un échec et renvoie une erreur de privilège.

### Bacs à sable utilisés

Un bac à sable répondant aux critères de suppression mais actuellement utilisé n'est pas supprimé. Une entrée apparaît dans le journal de débogage au niveau informations pour indiquer l'occurrence.

### Dates d'accès et de mise à jour

Les attributs de date peuvent être des correspondances avec au plus le détail des jours. La collecte de ces attributs est donc limitée en raison de cette restriction. La date de la dernière mise à jour n'est pas actualisée lors d'écritures de cellule individuelles. A la place, le système sauvegarde la date de déchargement du bac à sable dans lequel ces opérations ont été réalisées pendant qu'il était en mémoire. Pour ce bac à sable, la date du dernier accès et de la dernière mise à jour sont identiques. Seule la date du dernier accès est mise à jour lorsqu'un bac à sable est déchargé de la mémoire. Les bacs à sable en mémoire n'étant par ailleurs pas soumis à `ServerSandboxesDelete`, la date du dernier accès n'est pas mise à jour lorsqu'un bac à sable est chargé en mémoire.

Prenons le scénario d'utilisation suivant :

Tableau 2. Exemple - Date du dernier accès		
Jour	Période	Action
1	1	Charger le bac à sable S
1	2	Ecrire 1
2	3	Lire 1
2	4	Décharger le bac à sable

Un utilisateur travaille avec un bac à sable pendant deux jours (peut-être pendant une période beaucoup plus courte du fait du changement de jour). A la période 4, lors du déchargement du bac à sable, la date de la dernière mise à jour prend la valeur 2 et non 1, qui correspond pourtant à la date réelle de la dernière mise à jour. La date du dernier accès prend également la valeur 2 pour la période 4. Si l'écriture de 1 avait été une lecture, seule la date du dernier accès aurait pris la valeur 2, alors que la date de la dernière mise à jour n'aurait pas changé.

### Exemple

```
ServerSandboxesDelete( 'client:=Admin, name:=:best case scenario' );
```

## ServerSandboxDiscardAllChanges

`ServerSandboxDiscardAllChanges` supprime toutes les modifications dans un bac à sable existant.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxDiscardAllChanges( nom_bac_à_sable );
```

Argument	Description
<code>nom_bac_à_sable</code>	Valeur de chaîne. Nom d'un bac à sable appartenant à l'utilisateur qui exécute la fonction.

### Exemple

```
ServerSandboxDiscardAllChanges( 'MySandbox' );
```

## ServerSandboxMerge

ServerSandboxMerge fusionne un bac à sable source avec un bac à sable cible existant. Si le bac à sable cible n'est pas spécifié, le bac à sable source est fusionné avec les données de base.

Les bacs à sable sont des espaces de travail privés dans lesquels un utilisateur peut saisir et stocker des valeurs de données, séparément des données de base de TM1. Les bacs à sable sont stockés sur le disque et, lorsqu'ils sont utilisées, en mémoire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxMerge( srce, cible, conflictRes, waitForLocks);
```

Argument	Description
src	Nom du bac à sable source dont l'utilisateur actif est propriétaire et qui doit être fusionné avec le bac à sable < <i>cible</i> >.  Le bac à sable < <i>srce</i> > n'est pas modifié.  Requis String
cible	Nom d'un bac à sable dont l'utilisateur actif est propriétaire et qui doit être fusionné avec le bac à sable < <i>srce</i> >.  Le bac à sable < <i>cible</i> > est mis à jour.  Si < <i>cible</i> > est vide, vous fusionnez < <i>srce</i> > avec les données de base et mettez à jour les données de base.  Obligatoire. Pour laisser ce paramètre vide, utilisez deux apostrophes concaténées : ' ' .  String
conflictRes	Le paramètre < <i>conflictRes</i> > est ignoré.  Facultatif Numérique
waitForLocks	Le paramètre < <i>waitForlocks</i> > est un entier qui indique si le processus doit attendre l'application de verrous pour garantir la sérialisation.  1 signifie que le processus doit attendre l'application de verrous, intercepter toutes les exceptions de conflit et effectuer une nouvelle tentative au lieu de permettre l'annulation de la tâche ou du processus.  0 signifie qu'il n'est pas nécessaire d'attendre l'application de verrous. Les exceptions entraînant une annulation sont autorisées.  Facultatif Numérique

### Exemple

Fusion de mySandbox dans les données de base.

```
ServerSandboxMerge(mySandbox, '');
```

## ServerSandboxExists

ServerSandboxExists vérifie l'existence du bac à sable indiqué. La valeur 1 est renvoyée si le bac à sable existe ; sinon, la valeur 0 est renvoyée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxExists( sandboxname )
```

ou

```
ServerSandboxExists( sandboxname , username )
```

### Arguments

Nom du bac à sable dont l'existence est testée. ServerSandboxExists prend la valeur d'un paramètre de chaîne facultatif (nom du client propriétaire). Le client appelant peut utiliser le paramètre facultatif pour spécifier un client autre que lui-même, s'il dispose des droits appropriés. Une erreur liée aux droits est renvoyée si le client spécifié n'est pas l'exécutant et si le client exécutant n'est pas membre du groupe DataAdmin ou ADMIN. Si le paramètre facultatif n'est pas utilisé, les bacs à sable du client actif constituent le sujet.

### Exemple

Le fragment suivant montre comment les fonctions ServerSandboxExists, ServerSandboxGet, et ServerSandboxListCountGet peuvent être utilisées pour répéter les bacs à sable de l'utilisateur User1 et rediriger ces bacs à sable vers un fichier texte. Le processus TurboIntegrator s'exécute pour les membres des groupes Admin et Data Admin et pour l'utilisateur User1. Le processus TurboIntegrator échoue et renvoie une erreur liée aux droits pour tous les autres utilisateurs.

```
SandboxIndex = 1;
NumSandboxes = ServerSandboxListCountGet( 'User1' );

WHILE( SandboxIndex <= NumSandboxes );

    SandboxName = ServerSandboxGet( SandboxIndex, 'User1' );
    IF( ServerSandboxExists( SandboxName, 'User1' ) = 1 );
        ASCIIOUTPUT( 'C:\User1Sandboxes.txt', SandboxName );
    ENDIF;

    SandboxIndex = SandboxIndex + 1;

END;
```

## ServerSandboxGet

ServerSandboxGet renvoie le nom du bac à sable identifié par le nombre *N*, où *N* correspond au paramètre entré.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerSandboxGet( index )
```

ou

```
ServerSandboxGet( index, username )
```

## Arguments

Index du bac à sable demandé dans la collection de bacs à sable de l'utilisateur. L'espace index doit être continu, de sorte que la première occurrence de chaîne vide renvoyée puisse être utilisée pour interrompre l'itération. La suppression d'un bac à sable modifie les index des autres bacs de la liste.

`ServerSandboxGet` prend la valeur d'un paramètre de chaîne facultatif (nom du client propriétaire). Le client appelant peut utiliser le paramètre facultatif pour spécifier un client autre que lui-même, s'il dispose des droits appropriés. Une erreur liée aux droits est renvoyée si le client spécifié n'est pas l'exécutant et si le client exécutant n'est pas membre du groupe DataAdmin ou ADMIN. Si le paramètre facultatif n'est pas utilisé, les bacs à sable du client actif constituent le sujet.

## Exemple

Le fragment suivant montre comment les fonctions `ServerSandboxExists`, `ServerSandboxGet`, et `ServerSandboxListCountGet` peuvent être utilisées pour répéter les bacs à sable de l'utilisateur `User1` et rediriger ces bacs à sable vers un fichier texte. Le processus `TurboIntegrator` s'exécute pour les membres des groupes Admin et Data Admin et pour l'utilisateur `User1`. Le processus `TurboIntegrator` échoue et renvoie une erreur liée aux droits pour tous les autres utilisateurs.

```
SandboxIndex = 1;
NumSandboxes = ServerSandboxListCountGet( 'User1' );

WHILE( SandboxIndex <= NumSandboxes );

    SandboxName = ServerSandboxGet( SandboxIndex, 'User1' );

    IF( ServerSandboxExists( SandboxName, 'User1' ) = 1 );

        ASCIIOUTPUT( 'C:\User1Sandboxes.txt', SandboxName );

    ENDIF;

    SandboxIndex = SandboxIndex + 1;

END;
```

## ServerSandboxListCountGet

`ServerSandboxListCountGet` renvoie le nombre de bacs à sable sous forme de valeur numérique.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ServerSandboxListCountGet()
```

ou

```
ServerSandboxListCountGet( username )
```

## Arguments

`ServerSandboxListCountGet` prend la valeur d'un paramètre de chaîne facultatif (nom du client propriétaire). Le client appelant peut utiliser le paramètre facultatif pour spécifier un client autre que lui-même, s'il dispose des droits appropriés. Une erreur liée aux droits est renvoyée si le client spécifié n'est pas l'exécutant et si le client exécutant n'est pas membre du groupe DataAdmin ou ADMIN. Si le paramètre facultatif n'est pas utilisé, les bacs à sable du client actif constituent le sujet.

## Exemple

Le fragment suivant montre comment les fonctions `ServerSandboxExists`, `ServerSandboxGet`, et `ServerSandboxListCountGet` peuvent être utilisées pour répéter les bacs à sable de l'utilisateur `User1` et rediriger ces bacs à sable vers un fichier texte. Le processus `TurboIntegrator` s'exécute pour les

membres des groupes Admin et Data Admin et pour l'utilisateur User1. Le processus TurboIntegrator échoue et renvoie une erreur liée aux droits pour tous les autres utilisateurs.

```
SandboxIndex = 1;
NumSandboxes = ServerSandboxListCountGet( 'User1' );

WHILE( SandboxIndex <= NumSandboxes );

    SandboxName = ServerSandboxGet( SandboxIndex, 'User1' );

    IF( ServerSandboxExists( SandboxName, 'User1' ) = 1 );

        ASCIIOUTPUT( 'C:\User1Sandboxes.txt', SandboxName );

    ENDIF;

    SandboxIndex = SandboxIndex + 1;

END;
```

## SetUseActiveSandboxProperty

SetUseActiveSandboxProperty indique si un processus doit lire et écrire les données de cube dans les données de base ou dans le bac à sable actif de l'utilisateur. Par défaut, les processus lisent et écrivent dans les données de base.

La portée de cette fonction s'applique uniquement au processus en cours d'exécution et remplace temporairement la valeur permanente de cette propriété, définie dans l'interface utilisateur de l'Explorateur de serveur / Architect.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SetUseActiveSandboxProperty(PropertyValue)
```

Argument	Description
PropertyValue	Valeur booléenne indiquant si le processus doit utiliser le contexte du bac à sable actif lors de la lecture et de l'écriture des données de cube.  Si PropertyValue = 0, le processus ignore le contexte du bac à sable actif et lit/écrit les données dans les données de base.  Si PropertyValue = 1, le processus lit/écrit les données de cube dans le bac à sable actif.

### Exemple

```
SetUseActiveSandboxProperty(1);
```

Dans cet exemple, le processus lira/écrira les données de cube dans le bac à sable actif pour le reste de cette exécution.

## Fonctions TurboIntegrator de sécurité

Ces fonctions sont utilisées pour la sécurité.

## AddClient

AddClient crée un client sur le serveur. Les modifications apportées par la fonction AddClient ne sont appliquées que lorsque la procédure Métadonnées d'un processus est terminée. Cette fonction, comme toutes les fonctions qui mettent à jour des métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AddClient(ClientName);
```

Argument	Description
ClientName	Nom du client à ajouter au serveur. Le nom du client est limité à 255 caractères/octets.

### Exemple

```
AddClient('Brian');
```

Cet exemple ajoute le client Brian au serveur.

## AddGroup

AddGroup crée un groupe d'utilisateurs sur le serveur. Les modifications apportées par la fonction AddGroup ne sont appliquées que lorsque la procédure Métadonnées d'un processus est terminée. Cette fonction, comme toutes les fonctions qui mettent à jour des métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AddGroup(GroupName);
```

Argument	Description
GroupName	Nom du groupe à créer.

### Exemple

```
AddGroup('Finances');
```

Cette fonction ajoute le groupe d'utilisateurs Finances au serveur.

## AssignClientToGroup

AssignClientToGroup affecte un client existant sur un serveur à un groupe d'utilisateurs existant. Cette fonction affecte un client existant sur un serveur à un groupe d'utilisateurs existant.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AssignClientToGroup(ClientName, GroupName);
```

Argument	Description
ClientName	Nom du client à affecter à un groupe
GroupName	Groupe auquel vous voulez affecter le client.

### Exemple

```
AssignClientToGroup('Brian', 'Finances');
```

Cet exemple affecte le client existant Brian au groupe d'utilisateurs existant Finances.

### AssignClientPassword

AssignClientPassword affecte un mot de passe à un client existant sur un serveur. AssignClientPassword renvoie 1 si l'affectation de mot de passe réussit et 0 si elle échoue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AssignClientPassword (ClientName, Password);
```

Argument	Description
ClientName	Nom du client pour lequel vous voulez affecter un mot de passe.
Password	Mot de passe à affecter au client. Utilisez du texte en clair lorsque vous affectez un mot de passe. TM1 chiffrera le mot de passe sur le serveur.  Les mots de passe doivent avoir au moins 5 caractères de longueur.

### Exemple

```
AssignClientPassword ('Brian', 'marinier');
```

Cet exemple affecte le mot de passe 'marinier' au client nommé Brian.

### AssociateCAMIDToGroup

AssociateCAMIDToGroup crée une association entre un groupe d'utilisateurs TM1 et un groupe CAMID.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AssociateCAMIDToGroup(GroupName, CAMID, CAMIDDisplayValue);
```

Argument	Description
GroupName	Nom du groupe TM1 auquel vous voulez associer le groupe CAMID.



Argument	Description
CAMID	Nom du groupe CAMID. Si le groupe CAMID n'existe pas, il sera créé dans le cube de contrôle }ClientCAMAssociatedGroups.
CAMIDDefDisplayValue	Alias du groupe CAMID.

## CellSecurityCubeCreate

CellSecurityCubeCreate crée un cube de sécurité à partir d'un cube existant à l'aide d'un ensemble de dimensions réduit. Cette fonction, comme toutes les fonctions qui mettent à jour des métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellSecurityCubeCreate ('DataCube', '0:0:1:0');
```

Argument	Description
Cube	Nom du cube de données.
DimensionMap	Chaîne spécifiant si la dimension à chaque position doit être utilisée dans le cube de sécurité. L'ordre des dimensions est celui du cube d'origine. Un 1 pour chaque dimension incluse et un 0 pour une dimension exclue. Chaque valeur est séparée par un point.
Booléen	La valeur est True si l'opération a abouti. Sinon, erreur majeure.
Informations supplémentaires	La propriété GrantSecurityAccess doit être définie pour que ce processus TurboIntegrator aboutisse. Crée le cube de sécurité de cellule.

### Exemple

```
CellSecurityCubeCreate ('DataCube', '0:0:1:0');
```

Cet exemple crée un cube RDCLS à partir du cube appelé Data Cube.

## CellSecurityCubeDestroy

CellSecurityCubeDestroy supprime un cube de sécurité qui a été créé à partir d'un cube existant. Cette fonction, comme toutes les fonctions qui mettent à jour des métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
CellSecurityCubeDestroy ('DataCube', '0:0:1:0');
```

Argument	Description
Cube	Nom du cube de données.
Booléen	La valeur est True si l'opération a abouti. Sinon, erreur majeure.
Informations supplémentaires	La propriété GrantSecurityAccess doit être définie pour que ce processus TurboIntegrator aboutisse. Détruit le cube de sécurité de cellule.

### Exemple

```
CellSecurityCubeDestroy ('DataCube');
```

## DeleteClient

DeleteClient supprime un client du serveur. Les changements appliqués au moyen de la fonction DeleteClient ne sont mis en application que lorsque la procédure Métadonnées d'un processus est complétée. Cette fonction, comme toutes les fonctions qui mettent à jour les métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DeleteClient(ClientName);
```

Argument	Description
ClientName	Nom du client à supprimer du serveur.

### Exemple

```
DeleteClient('Brian');
```

Cet exemple supprime le client Brian du serveur.

## DeleteGroup

DeleteGroup supprime un groupe d'utilisateurs du serveur. Les modifications apportées par la fonction DeleteGroup ne sont appliquées que lorsque la procédure Métadonnées d'un processus est terminée. Cette fonction, comme toutes les fonctions qui mettent à jour des métadonnées, ne doit pas être utilisée dans les onglets Données ou Epilogue d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
DeleteGroup(GroupName);
```

Argument	Description
GroupName	Groupe à supprimer.

## Exemple

```
DeleteGroup('Finances');
```

Cette fonction supprime le groupe d'utilisateurs Finances du serveur.

## ElementSecurityGet

ElementSecurityGet extrait le niveau de sécurité affecté à un groupe spécifié pour un élément de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ElementSecurityGet(DimName, ElName, Group);
```

Argument	Description
DimName	Dimension parent de l'élément dont vous récupérez le niveau de sécurité.
ElName	Élément dont vous récupérez le niveau de sécurité.
Group	Groupe d'utilisateurs pour lequel vous récupérez un niveau de sécurité.

## Exemple

```
ElementSecurityGet('Region', 'Germany', 'Budgeting');
```

Cet exemple renvoie le niveau de sécurité affecté au groupe Budgeting pour l'élément Germany de la dimension Region.

## ElementSecurityPut

ElementSecurityPut affecte un niveau de sécurité à un groupe spécifié pour un élément de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ElementSecurityPut(Level, DimName, ElName, Group);
```

Argument	Description
Level	Niveau de sécurité que vous affectez. Il existe six valeurs de niveau possibles : <ul style="list-style-type: none"><li>• Aucun</li><li>• Lecture</li><li>• Ecriture</li><li>• Réservation</li><li>• Lock</li><li>• Admin</li></ul>

Argument	Description
DimName	Dimension parent de l'élément auquel vous affectez un niveau de sécurité.
ElName	Élément auquel vous affectez un niveau de sécurité.
Group	Groupe d'utilisateurs auquel vous affectez un niveau de sécurité.

### Exemple

```
ElementSecurityPut('Réserve', 'Région', 'Allemagne', 'Budget');
```

Cet exemple affecte le niveau Reserve au groupe Budgeting pour l'élément Germany de la dimension Region.

### HierarchyElementSecurityGet

HierarchyElementSecurityGet extrait le niveau de sécurité affecté à un groupe spécifié pour un élément de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchyElementSecurityGet(DimName, HierName, ElName, Group);
```

Argument	Description
DimName	Dimension parent de l'élément dont vous récupérez le niveau de sécurité.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément dont vous récupérez le niveau de sécurité.
Group	Groupe d'utilisateurs pour lequel vous récupérez un niveau de sécurité.

### Exemple

```
HierarchyElementSecurityGet('Region', 'Europe', 'Germany', 'Budgeting');
```

Cet exemple renvoie le niveau de sécurité affecté au groupe d'utilisateurs Budgeting de l'élément Germany. L'élément apparaît dans la hiérarchie Europe de la dimension Region.

### HierarchyElementSecurityPut

HierarchyElementSecurityPut affecte un niveau de sécurité à un groupe spécifié pour un élément de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchyElementSecurityPut(Level, DimName, HierName, ElName, Group);
```

Argument	Description
Level	Niveau de sécurité que vous affectez. Il existe six valeurs de niveau possibles : <ul style="list-style-type: none"><li>• Aucun</li><li>• Lecture</li><li>• Ecriture</li><li>• Réservation</li><li>• Lock</li><li>• Admin</li></ul>
DimName	Dimension parent de l'élément auquel vous affectez un niveau de sécurité.
HierName	Nom de la hiérarchie dans la dimension.
ElName	Élément auquel vous affectez un niveau de sécurité.
Group	Groupe d'utilisateurs auquel vous affectez un niveau de sécurité.

## Exemple

```
HierarchyElementSecurityPut('Reserve', 'Region', 'Europe', 'Germany', 'Budgeting');
```

Cet exemple affecte la sécurité Reserve au groupe Budgeting de l'élément Germany. L'élément apparaît dans la hiérarchie Europe de la dimension Region.

## RemoveCAMIDAssociation

RemoveCAMIDAssociation supprime toutes les associations entre les groupes d'utilisateurs TM1 et un groupe CAMID spécifié.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
RemoveCAMIDAssociation(CAMID, RemoveCAMID);
```

Argument	Description
CAMID	Nom du groupe CAMID pour lequel vous voulez supprimer toutes les associations de sécurité.

Argument	Description
RemoveCAMID	Détermine si le groupe CAMID spécifié est supprimé du cube de contrôle }ClientCAMAssociatedGroups.  La valeur 0 permet de conserver le groupe CAMID dans le cube de contrôle }ClientCAMAssociatedGroups.  La valeur 1 permet de supprimer le groupe CAMID du cube de contrôle }ClientCAMAssociatedGroups.

### RemoveCAMIDAssociationFromGroup

RemoveCAMIDAssociationFromGroup supprime une association entre un groupe d'utilisateurs TM1 et un groupe CAMID.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

```
RemoveCAMIDAssociationFromGroup(Group Name, CAMID);
```

Argument	Description
Group Name	Nom du groupe d'utilisateurs TM1 pour lequel vous voulez supprimer l'association.
CAMID	Nom du groupe d'utilisateurs CAMID pour lequel vous voulez supprimer l'association.

### RemoveClientFromGroup

RemoveClientFromGroup supprime un client spécifié d'un groupe d'utilisateurs.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

#### Syntaxe

```
RemoveClientFromGroup(Client Name, Group Name);
```

Argument	Description
Client Name	Client à supprimer
Group Name	Groupe d'utilisateurs dont vous voulez supprimer le client.

#### Exemple

```
RemoveClientFromGroup('Brian', 'Finances');
```

Cet exemple supprime le client Brian du groupe d'utilisateurs Finances.

## SetHierarchyGroupsSecurity

SetHierarchyGroupsSecurity définit le niveau de sécurité de tous les groupes existants pour la hiérarchie de dimension spécifiée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SetHierarchyGroupsSecurity(securityLevel, dimension, hierarchy)
```

Argument	Description
securityLevel	Niveau de sécurité que vous affectez. Il existe six valeurs possibles : <ul style="list-style-type: none"><li>• Aucun</li><li>• Lecture</li><li>• Ecriture</li><li>• Réservation</li><li>• Lock</li><li>• Admin</li></ul>
dimension	Nom de la dimension.
hierarchy	Nom de la hiérarchie de dimension.

### Exemple

```
SetHierarchyGroupsSecurity('Reserve', 'Region', 'Europe');
```

Cet exemple affecte la sécurité Reserve à tous les groupes existants dans la hiérarchie Europe de la dimension Region.

## SetHierarchyElementGroupsSecurity

SetHierarchyElementGroupsSecurity définit le niveau de sécurité d'un élément spécifié d'une hiérarchie dans une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SetHierarchyElementGroupsSecurity(securityLevel, dimension, hierarchy, element)
```

Argument	Description
securityLevel	Niveau de sécurité que vous affectez. Il existe six valeurs possibles : <ul style="list-style-type: none"><li>• Aucun</li><li>• Lecture</li><li>• Ecriture</li><li>• Réservation</li><li>• Lock</li><li>• Admin</li></ul>

Argument	Description
dimension	Nom de la dimension.
hierarchy	Nom de la hiérarchie de dimension.
element	Élément auquel vous affectez un niveau de sécurité.

### Exemple

```
SetHierarchyElementGroupsSecurity('Reserve', 'Region', 'Europe', 'Germany');
```

Cet exemple affecte la sécurité Reserve à l'élément Germany de la hiérarchie Europe de la dimension Region.

### SetDimensionGroupsSecurity

SubsetExpandAboveSet définit le niveau de sécurité de tous les groupes existants pour la dimension spécifiée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SetDimensionGroupsSecurity(securityLevel, dimension)
```

Argument	Description
securityLevel	Niveau de sécurité que vous affectez. Il existe six valeurs possibles : <ul style="list-style-type: none"> <li>• Aucun</li> <li>• Lecture</li> <li>• Ecriture</li> <li>• Réservation</li> <li>• Lock</li> <li>• Admin</li> </ul>
dimension	Nom de la dimension.

### Exemple

```
SetDimensionGroupsSecurity('Reserve', 'Region');
```

Cet exemple affecte la sécurité Reserve à tous les groupes existants de la dimension Region.

### SetElementGroupsSecurity

SetElementGroupsSecurity définit le niveau de sécurité d'un élément spécifié dans une dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SetElementGroupsSecurity(securityLevel, dimension, element)
```



Argument	Description
securityLevel	Niveau de sécurité que vous affectez. Il existe six valeurs possibles : <ul style="list-style-type: none"> <li>• Aucun</li> <li>• Lecture</li> <li>• Ecriture</li> <li>• Réservation</li> <li>• Lock</li> <li>• Admin</li> </ul>
dimension	Nom de la dimension.
element	Elément auquel vous affectez un niveau de sécurité.

### Exemple

```
SetElementGroupsSecurity('Reserve', 'Region', 'Germany');
```

Cet exemple affecte la sécurité Reserve à l'élément Germany de la dimension Region.

## SecurityOverlayGlobalLockCell

SecurityOverlayGlobalLockCell est utilisé pour limiter à la lecture seule les droits d'accès d'un noeud en le verrouillant. Parce qu'elle utilise la surcouche globale, elle affecte tous les utilisateurs. Le cube de surcouche doit avoir été créé avant l'utilisation de la commande. Les éléments fournis dans l'adresse doivent être limités aux dimensions utilisées dans la surcouche.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SecurityOverlayGlobalLockCell(bLock, Cube, element1,..., elementN)
```

Argument	Description
bLock	1 le verrouille. 0 le déverrouille.
Cube	Nom du cube.
elementN	Nom de l'élément de surcouche qui définit le tuple. L'ordre doit correspondre à l'ordre des dimensions d'origine du cube.
Booléen	La valeur est True si l'opération a abouti. Sinon, erreur majeure.
Informations supplémentaires	La propriété GrantSecurityAccess doit être définie pour que ce processus TurboIntegrator aboutisse. Crée le cube de surcouche de sécurité globale par défaut. Les surcouches globales s'appliquent à tous les utilisateurs.

## Exemple

```
SecurityOverlayGlobalLockCell(1,'Sales','MA');  
SecurityOverlayGlobalLockCell(0,'Products','MA','2011');
```

Dans le premier exemple, une seule dimension est utilisée pour la surcouche. Le deuxième exemple utilise deux dimensions.

## SecurityOverlayCreateGlobalDefault

SecurityOverlayCreateGlobalDefault est utilisé pour créer ou supprimer un cube de surcouche de sécurité et pour définir la surcouche pour une zone particulière d'un cube de données.

La création d'un cube de données avec un nom signifie qu'un cube de surcouche convertit le cube de données en surcouche au redémarrage du serveur. Au moment de son chargement, le cube est configuré en surcouche si un cube de données correspondant est détecté.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SecurityOverlayCreateGlobalDefault (Cube,  
DimensionMap)
```

Argument	Description
Cube	Nom du cube.
DimensionMap	Chaîne spécifiant si la dimension à chaque position doit être utilisée dans la surcouche. L'ordre des dimensions est celui du cube d'origine. Un 1 pour chaque dimension incluse et un 0 pour une dimension exclue. Chaque valeur est séparée par un point.
Booléen	La valeur est True si l'opération a abouti. Sinon, erreur majeure.
Informations supplémentaires	La propriété GrantSecurityAccess doit être définie pour que ce processus TurboIntegrator aboutisse. Crée le cube de surcouche de sécurité globale par défaut. Les surcouches globales s'appliquent à tous les utilisateurs.

## Exemple

```
SecurityOverlayCreateGlobalDefault('DataCube',  
'0:0:1:0');
```

## SecurityOverlayDestroyGlobalDefault

La fonction SecurityOverlayDestroyGlobalDefault est utilisée pour supprimer un cube de surcouche de sécurité et pour définir la surcouche pour une zone particulière d'un cube de données.

La création d'un cube de données avec un nom signifie qu'un cube de surcouche convertit le cube de données en surcouche au redémarrage du serveur. Au moment de son chargement, le cube est configuré en surcouche si un cube de données correspondant est détecté.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.



Argument	Description
Informations supplémentaires	La propriété GrantSecurityAccess doit être définie pour que ce processus TurboIntegrator aboutisse. Crée le cube de surcouche de sécurité globale par défaut. Les surcouches globales s'appliquent à tous les utilisateurs.

### Exemples

```
SecurityOverlayGlobalLockNode(1, 'Sales', 'MA');
SecurityOverlayGlobalLockNode(0, 'Products', 'MA | 2011');
SecurityOverlayGlobalLockNode(0, 'Products', 'MA : 2011', ':');
```

Dans le premier exemple, une seule dimension est utilisée pour la surcouche. Les deux autres exemples utilisent deux dimensions.

## SecurityRefresh

SecurityRefresh lit tous les cubes de contrôle de sécurité et régénère les structures internes sur le serveur qui sont utilisées par les fonctions API de TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SecurityRefresh;
```

### Arguments

Aucun.

## Fonctions TurboIntegrator de manipulation de serveur

Ces fonctions facilitent la manipulation des serveurs.

### BatchUpdateFinish

BatchUpdateFinish demande au serveur de quitter le mode de mise à jour par lots.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

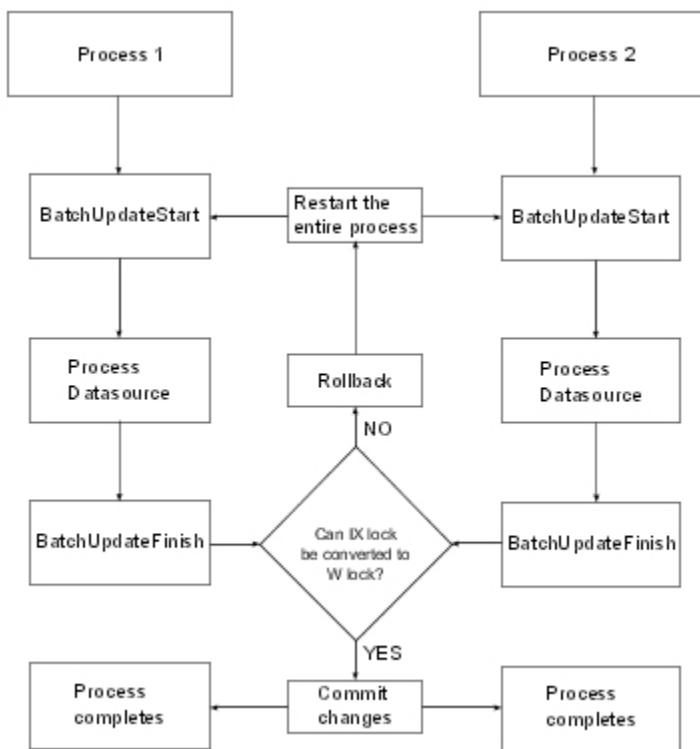
### Sémantique

Lorsque plusieurs processus s'exécutent en mode de mise à jour par lots et appliquent des modifications à un cube unique, le schéma de verrouillage de TM1 peut empêcher l'un des processus de mettre à jour le cube. Ce blocage est intentionnel : lorsque un processus obtient de verrouiller l'écriture de modifications dans un cube, les autres processus ne peuvent pas écrire dans ce cube, et ceci pour préserver l'intégrité des données.

Ce schéma de verrouillage peut être illustré en utilisant un exemple de deux processus, Processus 1 et Processus 2 qui mettent à jour un cube unique.

- Les deux processus démarrent et appellent la fonction BatchUpdateStart pour lancer les mises à jour par lots.
- Chaque processus est exécuté sur une source de données unique.
- Processus 1 effectue le traitement des données et appelle la fonction BatchUpdateFinish. Le processus obtient un verrouillage de l'écriture dans le cube et sauvegarde les modifications.

- Pendant que Processus 1 verrouille toujours l'écriture dans le cube, Processus 2 termine le traitement des données et appelle la fonction BatchUpdateFinish. Cependant, puisque Processus 1 continue à verrouiller l'écriture, Processus 2 ne peut pas verrouiller le cube. Toutes les modifications de données appliquées dans Processus 2 sont annulées et Processus 2 est relancé. Ceci assure l'intégrité des données.



En fonction de la taille de la source de données pour Processus 2, l'annulation des données et la réexécution du processus peuvent entraîner une diminution sensible de la performance. Pour résoudre ce problème, pensez à utiliser la fonction [BatchUpdateFinishWait](#) à la place de BatchUpdateFinish.

### Syntaxe

```
BatchUpdateFinish(SaveChanges);
```

Argument	Description
SaveChanges	Cet argument est un signal ordonnant au serveur de sauvegarder ou d'abandonner les modifications faites en mode de mise à jour par lots.  Spécifiez 0 pour sauvegarder les modifications, 1 pour les abandonner.

### Exemple

```
BatchUpdateFinish(0);
```

Cet exemple indique au serveur de sauvegarder les modifications des données TM1 et de quitter le mode de mise à jour par lots.

### BatchUpdateFinishWait

BatchUpdateFinishWait est identique à BatchUpdateFinish, sauf que le processus attend que le verrou soit disponible et valide ensuite les modifications. Si un processus appelle BatchUpdateFinishWait mais

ne parvient pas sécuriser un verrou en écriture pour le cube afin de valider les modifications, le processus attend que le verrou soit disponible, puis valide les modifications.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

Les modifications de données appliquées dans le processus ne sont pas annulées et le processus n'est pas réexécuté.

**Remarque :** En attendant le verrou en écriture du cube, le processus libère tous les verrous en lecture qu'il avait obtenus pour d'autres objets pendant son exécution. Etant donné que ces verrous en lecture sont libérés avant que le processus ne puisse valider les modifications apportées au cube, il se peut que les objets dont les verrous en lecture sont libérés soient modifiés *avant* que le cube n'ait été mis à jour. Il peut en résulter des incohérences dans les données lors de l'utilisation de BatchUpdateFinishWait.

Nous recommandons d'utiliser BatchUpdateFinishWait uniquement dans des situations contrôlées dans lesquelles vous savez que d'autres processus ne modifient pas les données ou métadonnées relatives au processus appelant BatchUpdateFinishWait.

## Syntaxe

```
BatchUpdateFinishWait(SaveChanges);
```

Argument	Description
SaveChanges	Cet argument est un signal ordonnant au serveur de sauvegarder ou d'abandonner les modifications faites en mode de mise à jour par lots. Spécifiez 0 pour sauvegarder les modifications, 1 pour les abandonner.

## Exemple

```
BatchUpdateFinishWait(0);
```

Cet exemple indique au serveur de sauvegarder les modifications des données TM1 et de quitter le mode de mise à jour par lots.

## BatchUpdateStart

BatchUpdateStart active les mises à jour par lots.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
BatchUpdateStart;
```

## Arguments

Aucun.

## DisableBulkLoadMode

DisableBulkLoadMode désactive le traitement du chargement en bloc.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

Pour plus de détails, voir [«EnableBulkLoadMode»](#), à la page 375.

## EnableBulkLoadMode

EnableBulkLoadMode active le traitement du chargement en bloc pour un processus TurboIntegrator. Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

Vous pouvez activer le mode de chargement en bloc dans la section Prologue ou Epilogue d'un processus TurboIntegrator. Par souci d'efficacité, activez le mode de chargement en bloc dans la première instruction (ou du moins le plus près possible) de la section Prologue de votre processus.

Une fois le mode de chargement en bloc activé dans un processus, il ne peut être désactivé que sur la dernière ligne de la section Epilogue. Si vous tentez de désactiver le mode de chargement en bloc à tout autre instant du processus, la compilation n'aura pas lieu.

Si le mode est activé dans un processus TurboIntegrator, il reste activé tant qu'il n'est pas explicitement désactivé ou tant que la tâche n'est pas terminée. Autrement dit, vous pouvez activer le mode dans un processus au sein d'une tâche, puis exécuter toute une série de processus TurboIntegrator avant de le désactiver. Vous pouvez également activer et quitter le mode de chargement en bloc à plusieurs reprises, de façon à ne l'utiliser que pour certaines parties critiques d'une tâche.

Utilisez les commandes TurboIntegrator suivantes pour activer et désactiver le mode de chargement en bloc dans un processus TurboIntegrator.

```
EnableBulkLoadMode();
```

Utilisez la fonction TurboIntegrator suivante uniquement sur la dernière ligne de la section Epilogue de votre processus TI lorsque vous utilisez le mode de chargement en bloc.

```
DisableBulkLoadMode();
```

## RefreshMdxHierarchy

RefreshMdxHierarchy met à jour les hiérarchies MDX d'un serveur sans avoir à redémarrer le serveur.

Utilisez cette fonction après avoir configuré ou modifié les niveaux de hiérarchie nommés personnalisés d'une dimension dans le cube de contrôle }HierarchyProperties.

Pour plus d'informations sur l'utilisation de niveaux nommés avec les dimensions, reportez-vous à la section correspondante dans la documentation IBM Cognos *TM1 for Developers*.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
RefreshMdxHierarchy(dimensionName, hierarchy)
```

Argument	Description
dimensionName	Paramètre de chaîne facultatif indiquant une dimension spécifique à mettre à jour.  Laissez ce paramètre vide si vous souhaitez mettre à jour toutes les dimensions.
hierarchy	Nom de la hiérarchie dans la dimension. Ce paramètre est facultatif.

## Exemple

Mise à jour de toutes les dimensions :

```
RefreshMdxHierarchy('');
```

Pour mettre à jour uniquement la dimension Customers :

```
RefreshMdxHierarchy('customers');
```

## SaveDataAll

SaveDataAll sauvegarde toutes les données TM1 de la mémoire du serveur sur le disque et redémarre le fichier journal.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Utilisation de SaveDataAll dans une tâche

SaveDataAll sauvegarde toutes les modifications faites par une tâche avant d'appeler la fonction SaveDataAll.

Lorsqu'une tâche est en cours d'exécution, elle accumule les verrous sur les objets auxquels elle accède. L'opération d'enregistrement lancée par la fonction SaveDataAll supprime temporairement tous les verrous. Une fois l'enregistrement terminé, SaveDataAll rétablit tous les verrous précédents pour pouvoir continuer à accéder aux objets sur lesquels la fonction travaillait.

Il y a un bref laps de temps durant l'opération d'enregistrement pendant lequel les verrous sont levés et un autre utilisateur ou processus TurboIntegrator pourrait supprimer les objets utilisés par la tâche d'origine. Lorsque la tâche d'origine tente de récupérer les verrous sur ces objets, les objets ne sont pas disponibles et le traitement de la tâche s'arrête. Dans ce cas, une erreur similaire à la suivante est écrite dans le fichier Tm1s.log :

```
844 WARN 2008-04-01 16:40:09,734 TM1.Server TM1ServerImpl::FileSave could not reacquire lock on object with index 0x200002ca
```

### Contention de verrouillage et utilisation de SaveDataAll à la fin des processus TurboIntegrator

L'utilisation de SaveDataAll comme dernière commande d'un processus TurboIntegrator peut augmenter le conflit d'accès dans les processus TurboIntegrator TM1.

Dans les versions d'IBM TM1, la commande SaveDataAll était souvent ajoutée à la fin d'un processus TurboIntegrator chargeant des données avec la journalisation désactivée. SaveDataAll offrait un moyen d'écrire les données de la mémoire sur le disque directement après une importation, si bien que les données nouvellement importées n'étaient pas perdues en cas d'accident, comme une panne de serveur par exemple.

Toutefois, l'ajout de SaveDataAll en tant que dernière commande pouvait générer de nombreux processus d'importation TurboIntegrator se terminant chacun par la commande SaveDataAll. Cette technique fonctionnait dans TM1 version 9.0 et antérieures du fait du modèle de verrouillage précédent qui recourait uniquement au verrouillage en écriture global. Dans les versions précédentes, une seule opération en écriture pouvait avoir lieu à un moment donné. Par conséquent, la concurrence d'accès d'opérations SaveDataAll simultanées n'avait jamais lieu avec plusieurs opérations d'écriture simultanées.

La version 9.1 et versions ultérieures offrent un modèle de verrouillage par objet plus précis qui active les opérations d'écriture simultanées, si ces dernières ne rivalisent pas pour obtenir les mêmes ressources. Si c'est le cas, une contention de verrouillage se produit, forçant l'un des processus à annuler. Désormais, deux processus d'importation TurboIntegrator peuvent donc s'exécuter simultanément s'ils ne partagent aucun objet, par exemple s'ils importent dans deux cubes différents.

La fonction TurboIntegrator SaveDataAll s'appuie sur le fichier journal des transactions tm1s.log et fait appel à tous les objets contenus dans un modèle de données. Par conséquent deux processus



d'importation TurboIntegrator utilisant tous deux la fonction SaveDataAll, ne peuvent pas s'exécuter en parallèle : l'un sera exécuté, l'autre (et son processus TurboIntegrator) sera forcé à annuler. Le même principe vaut si les processus TurboIntegrator font partie des tâches : une seule tâche continuera d'exécuter la fonction TurboIntegrator SaveDataAll, l'autre tâche sera forcée à annuler.

Une annulation n'est pas souhaitable dans une optique de performance car elle augmente le temps d'exécution total d'un processus ou d'une tâche TurboIntegrator. La concurrence d'accès d'opérations SaveDataAll simultanées conduira toujours à une contention de verrouillage et à une annulation.

Il existe deux solutions pour éviter cette situation :

- N'utilisez pas la fonction TurboIntegrator SaveDataAll. Activez plutôt la journalisation des cubes d'importation.
- Si la journalisation des cubes d'importation ne peut pas être activée pour des raisons de performance, il ne doit y avoir qu'un processus appelant la fonction TurboIntegrator SaveDataAll dans l'application TM1. Utilisez une tâche distincte, unique et autonome pour exécuter l'opération SaveDataAll.

### Syntaxe

```
SaveDataAll;
```

### Arguments

Aucun.

## ServerShutdown

ServerShutdown arrête un serveur exécuté comme application. ServerShutdown ne peut pas être utilisée pour arrêter un serveur s'exécutant comme service Windows.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ServerShutDown(SaveData);
```

Argument	Description
SaveData	Valeur booléenne indiquant si le serveur doit sauvegarder les modifications sur disque avant l'arrêt.  Si SaveData = 0, les modifications ne sont pas enregistrées avant l'arrêt du serveur.  Si SaveData = 1, le serveur sauvegarde sur disque les modifications se trouvant en mémoire avant de s'arrêter.

### Exemple

```
ServerShutdown(1);
```

Cet exemple ferme le serveur et sauvegarde les données sur disque.

## Fonctions TurboIntegrator de manipulation de sous-ensembles

Ces fonctions facilitent la manipulation des sous-ensembles.

## HierarchySubsetAliasSet

La fonction HierarchySubsetAliasSet définit l'attribut d'alias à utiliser dans un sous-ensemble. Elle renvoie 1 si l'opération réussit ; sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetAliasSet( DimName, HierName, SubName, AliasName );
```

Argument	Description
DimName	Dimension parent du sous-ensemble pour lequel vous voulez définir l'alias.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Sous-ensemble pour lequel vous voulez définir l'alias.
AliasName	Alias à utiliser dans le sous-ensemble.

## HierarchySubsetCreate

HierarchySubsetCreate crée un sous-ensemble public vide pour une hiérarchie et une dimension spécifiées.

Lorsque l'argument facultatif AsTemporary est défini sur 1, le sous-ensemble est temporaire et n'est conservé que pendant la durée du processus ou de la tâche TurboIntegrator qui l'a créé.

### Remarque :

Dans TM1 Server version 11.2.0 et précédente, les sous-ensembles temporaires étaient visibles et pouvaient être utilisés uniquement par le processus qui les avait créés et par l'un de ses processus enfant. Les sous-ensembles temporaires n'étaient pas visibles pour les processus ancêtre et les processus apparentés. Vous pouviez créer des sous-ensembles portant un nom identique dans des processus enfant apparentés dotés du même processus parent.

Dans TM1 Server version 11.3.0 et suivante, ces sous-ensembles temporaires sont visibles pour les processus ancêtre et les processus apparentés. Si un processus TurboIntegrator parent "A" appelle deux processus TurboIntegrator enfant "A1" et "A2" et que le processus TurboIntegrator "A1" crée un sous-ensemble temporaire "S", le sous-ensemble temporaire "S" existe pendant la durée du processus parent TurboIntegrator "A". Vous ne pouvez pas créer un sous-ensemble temporaire portant le nom "S" dans le processus TurboIntegrator apparenté "A2" car le sous-ensemble est visible et utilisable par les processus apparentés "A1" et "A2".

Pendant la durée de son existence, un sous-ensemble temporaire est prioritaire par rapport à un sous-ensemble public du même nom. Si une autre fonction TurboIntegrator fait référence à un sous-ensemble qui existe à la fois à l'état temporaire et à l'état permanent, la fonction s'applique au sous-ensemble temporaire.

Aucun verrouillage n'est associé aux sous-ensembles temporaires, car ils ne sont jamais enregistrés. Cette particularité peut améliorer les performances, car TurboIntegrator n'a jamais besoin d'attendre la libération des verrous pour utiliser un sous-ensemble temporaire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetCreate(DimName, HierName, SubName, [AsTemporary]);
```

Argument	Description
DimName	Dimension parent du sous-ensemble que vous créez.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Nom à affecter au sous-ensemble.
AsTemporary	Argument facultatif qui indique si le sous-ensemble créé est temporaire. 1 indique un sous-ensemble temporaire, 0 indique un sous-ensemble permanent.  Si cet argument est omis, le sous-ensemble est permanent.

### Exemple

```
HierarchySubsetCreate('Region', 'European', 'Northern Europe', 1);
```

Cet exemple crée le sous-ensemble temporaire Northern Europe de la hiérarchie European dans la dimension Region. Vous pouvez maintenant ajouter des éléments au sous-ensemble.

### HierarchySubsetDeleteAllElements

HierarchySubsetDeleteAllElements supprime tous les éléments d'un sous-ensemble public d'une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetDeleteAllElements(DimName, HierName, SubsetName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez supprimer des éléments.
HierName	Nom de la hiérarchie dans la dimension.
SubsetName	Sous-ensemble dont vous voulez supprimer des éléments. Ce sous-ensemble doit être public. TurboIntegrator n'a pas accès aux objets privés.

### Exemple

```
oHierarchySubsetDeleteAllElements('Region', 'European', 'Central Europe');
```

Cet exemple supprime tous les éléments du sous-ensemble Europe centrale de la hiérarchie European dans la dimension Region.

### HierarchySubsetDestroy

HierarchySubsetDestroy supprime un sous-ensemble de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchySubsetDestroy(DimName, HierName, SubName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble que vous supprimez.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Nom du sous-ensemble à supprimer.

## Exemple

```
HierarchySubsetDestroy('Region', 'European', 'Northern Europe');
```

Cet exemple supprime le sous-ensemble Northern Europe de la hiérarchie European dans la dimension Region.

## HierarchySubsetElementExists

HierarchySubsetElementExists détermine si un élément spécifique existe dans un sous-ensemble public spécifique sur le serveur à partir duquel un processus TurboIntegrator est exécuté.

HierarchySubsetElementExists ne permet pas de déterminer si un élément existe dans un sous-ensemble privé.

Si l'élément existe dans le sous-ensemble spécifié, la fonction renvoie 1 ; sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchySubsetElementExists(DimName, HierName, SubsetName, ElementName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble contenant l'élément dont vous souhaitez confirmer l'existence.
HierName	Nom de la hiérarchie dans la dimension spécifiée.
SubsetName	Sous-ensemble public contenant l'élément dont vous souhaitez confirmer l'existence.
ElementName	Élément dont vous souhaitez confirmer l'existence.

## Exemple

```
HierarchySubsetElementExists('Region', 'Eastern', 'Europe', 'Italy');
```

Cet exemple détermine si l'élément Italy existe dans le sous-ensemble Europe de la hiérarchie Eastern de la dimension Region.

## HierarchySubsetElementDelete

HierarchySubsetElementDelete supprime un élément d'un sous-ensemble d'une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchySubsetElementDelete(DimName, HierName, SubName, Index);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez supprimer un élément.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Sous-ensemble dont vous voulez supprimer un élément.
Index	Numéro d'index de l'élément à supprimer du sous-ensemble.

## Exemple

```
HierarchySubsetElementDelete('Region', 'European', 'Northern Europe', 3);
```

Cet exemple supprime le troisième élément du sous-ensemble Northern Europe de la hiérarchie European dans la dimension Region.

## HierarchySubsetElementGetIndex

HierarchySubsetElementGetIndex extrait l'index d'un élément dans un sous-ensemble d'une hiérarchie de dimension.

La fonction renvoie l'index de la première occurrence de l'élément spécifié. Si l'élément n'existe pas dans le sous-ensemble ou qu'il est introuvable, la valeur zéro est renvoyée. Si la dimension ou le sous-ensemble est introuvable ou qu'un index de début hors plage est spécifié, une erreur est générée et la fonction TurboIntegrator est arrêtée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
HierarchySubsetElementGetIndex(DimName, HierName, SubsetName, ElementName, StartIndex);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
HierName	Nom de la hiérarchie dans la dimension.
SubsetName	Sous-ensemble qui contient l'élément.
ElementName	Nom d'élément à rechercher dans le sous-ensemble.
StartIndex	Numéro d'index à partir duquel la recherche doit commencer. La valeur doit être comprise entre 1 et la taille du sous-ensemble.

## Exemple

```
HierarchySubsetElementGetIndex('Region', 'Country', 'Europe', 'Italy', 3);
```

Cet exemple extrait l'index d'Italy du sous-ensemble Europe de la hiérarchie Country dans la dimension Region. La recherche commence à l'index 3.

## HierarchySubsetElementInsert

HierarchySubsetElementInsert ajoute un élément à un sous-ensemble existant d'une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetElementInsert(DimName, HierName, SubName, ElName, Position);
```

Argument	Description
DimName	Dimension parent du sous-ensemble auquel vous voulez ajouter un élément.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Nom du sous-ensemble auquel vous ajoutez un élément.
ElName	Nom de l'élément à ajouter au sous-ensemble. Cet élément doit exister dans la base de données TM1.
Position	Valeur indiquant la position d'index de l'élément dans le sous-ensemble.

### Exemple

```
HierarchySubsetElementInsert('Region', 'European', 'Northern Europe', 'Finland',3);
```

Cet exemple ajoute l'élément Finland au sous-ensemble Northern Europe de la hiérarchie European dans la dimension Region. Finland est le troisième élément dans la définition de sous-ensemble.

## HierarchySubsetExists

HierarchySubsetExists détermine si un sous-ensemble public spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. Cette fonction renvoie 1 si le sous-ensemble existe sur le serveur, sinon, elle renvoie 0. Notez que cette fonction ne peut pas être utilisée pour déterminer l'existence de sous-ensembles privés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetExists(DimName, HierName, SubsetName);
```

Argument	Description
DimName	Nom de la dimension parent du sous-ensemble dont vous voulez confirmer existence
HierName	Nom de la hiérarchie dans la dimension.
SubsetName	Nom du sous-ensemble public dont vous voulez confirmer l'existence.

### Exemple

```
HierarchySubsetExists('Region', 'Industrialized', 'Northern Europe');
```

Cet exemple détermine si le sous-ensemble Northern Europe existe dans la hiérarchie Industrialized de la dimension Region.

## HierarchySubsetGetSize

HierarchySubsetGetSize renvoie le nombre d'éléments d'un sous-ensemble d'une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetGetSize(DimName, HierName, SubsetName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez déterminer la taille.
HierName	Nom de la hiérarchie dans la dimension.
SubsetName	Sous-ensemble dont vous voulez déterminer la taille.

### Exemple

```
HierarchySubsetGetSize('Region', 'Eastern', 'EurAsia');
```

Cette fonction renvoie le nombre d'éléments du sous-ensemble EurAsia de la hiérarchie Eastern dans la dimension Region.

## HierarchySubsetGetElementName

HierarchySubsetGetElementName renvoie le nom de l'élément à un endroit spécifié de l'index dans un sous-ensemble donné d'une hiérarchie de dimension.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetGetElementName(DimName, HierName, SubsetName, ElementIndex);
```

Argument	Description
DimName	Parent du sous-ensemble depuis lequel vous voulez récupérer un nom d'élément.
HierName	Nom de la hiérarchie dans la dimension.
SubsetName	Sous-ensemble depuis lequel vous voulez récupérer un nom d'élément.
ElementIndex	Nombre représentant la position dans le sous-ensemble de l'élément à récupérer.

### Exemple

```
HierarchySubsetGetElementName('Region', 'Western', 'Americas', 4);
```

Cet exemple renvoie le quatrième élément du sous-ensemble Americas de la hiérarchie Western dans la dimension Region.

## HierarchySubsetIsAllSet

HierarchySubsetIsAllSet définit un sous-ensemble pour utiliser tous les éléments de la dimension parent. Cette fonction renvoie 1 si l'opération réussit ; sinon elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetIsAllSet( DimName, HierName, SubName, Flag );
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez utiliser tous les éléments.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Sous-ensemble dont vous voulez utiliser tous les éléments de dimension.
Flag	Une valeur non nulle indique que le sous-ensemble utilise tous les éléments en cours de la dimension parent et est mis à jour dynamiquement pour utiliser tous les éléments de la dimension parent chaque fois que le sous-ensemble est appelé.  La spécification d'une valeur zéro fige les éléments du sous ensemble comme étant l'ensemble en cours de tous les éléments de la dimension parent. A l'avenir, le sous-ensemble ne sera pas mis à jour dynamiquement pour utiliser tous les éléments de la dimension.

## HierarchySubsetMDXGet

HierarchySubsetMDXGet extrait l'expression MDX utilisée pour créer un sous-ensemble.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetMDXGet(DimName, HierName, SubName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Sous-ensemble pour lequel vous voulez extraire l'expression MDX.

### Exemple

```
mdxString = HierarchySubsetMDXGet('Cities', 'Italy', 'testsubset');
```

## HierarchySubsetMDXSet

HierarchySubsetMDXSet retire l'expression MDX d'un sous-ensemble dynamique et convertit le sous-ensemble dynamique en sous-ensemble statique sans verrouiller la dimension sous-jacente. Le contenu



du sous-ensemble est conservé mais l'expression est supprimée. Vous pouvez également remplir le sous-ensemble à l'aide d'une nouvelle expression MDX.

Tous les éléments qui existaient au moment de l'appel sont enregistrés en tant qu'éléments du nouveau sous-ensemble statique. Cette fonction renvoie le nombre d'éléments contenus dans le nouveau sous-ensemble.

**Remarque :** Cette fonction n'est exploitable qu'avec les sous-ensembles publics.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
HierarchySubsetMDXSet(DimName, HierName, SubName, MDX_expression);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
HierName	Nom de la hiérarchie dans la dimension.
SubName	Sous-ensemble pour lequel vous voulez retirer l'expression MDX.
MDX_expression	Expression MDX que vous souhaitez insérer dans le sous-ensemble. Tous les éléments sont supprimés et le sous-ensemble est rempli par les éléments définis par la nouvelle expression MDX. Si l'expression MDX est invalide, le traitement TurboIntegrator s'arrête, le contenu du sous-ensemble n'est pas modifié et une erreur est enregistrée.  Si vous n'utilisez pas ce paramètre, entrez une chaîne vide, comme illustré dans l'exemple.

### Exemple

```
countMem = HierarchySubsetMDXSet('Cities', 'Italy', 'testsubset', '');
```

### SubsetAliasSet

SubsetAliasSet définit l'attribut d'alias à utiliser dans un sous-ensemble. SubsetAliasSet renvoie 1 si la définition réussit, sinon elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetAliasSet( DimName, SubName, AliasName );
```

Argument	Description
DimName	Dimension parent du sous-ensemble pour lequel vous voulez définir l'alias.
SubName	Sous-ensemble pour lequel vous voulez définir l'alias.
AliasName	Alias à utiliser dans le sous-ensemble.

## SubsetCreate

SubsetCreate crée un sous-ensemble public vide pour une dimension spécifiée.

Lorsque l'argument `AsTemporary` a pour valeur 1, le sous-ensemble est temporaire et n'est conservé que pendant la durée du processus TurboIntegrator ou de la tâche à validation unique qui le crée. Si un processus TurboIntegrator parent appelle des processus TurboIntegrator enfant à l'aide de la fonction `ExecuteProcess` ou `ExecuteProcessWithReturn` et que le sous-ensemble temporaire est créé dans l'un de ces processus TurboIntegrator enfant, le sous-ensemble est conservé pendant la durée du processus TurboIntegrator parent.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Remarque :

Dans TM1 Server version 11.2.0 et précédente, les vues et les sous-ensembles temporaires étaient visibles et pouvaient être utilisés uniquement par le processus qui les avait créés et par l'un de ses processus enfant. Les vues et les sous-ensembles temporaires n'étaient pas visibles pour les processus ancêtre et les processus apparentés. Vous pouviez créer des sous-ensembles portant un nom identique dans des processus enfant apparentés dotés du même processus parent.

Dans TM1 Server version 11.3.0 et suivante, ces sous-ensembles temporaires sont visibles pour les processus ancêtre et les processus apparentés. Si un processus TurboIntegrator parent "A" appelle deux processus TurboIntegrator enfant "A1" et "A2" et que le processus TurboIntegrator "A1" crée un sous-ensemble temporaire "S", le sous-ensemble temporaire "S" existe pendant la durée du processus parent TurboIntegrator "A". Vous ne pouvez pas créer un sous-ensemble temporaire portant le nom "S" dans le processus TurboIntegrator apparenté "A2" car le sous-ensemble est visible et utilisable par les processus apparentés "A1" et "A2".

Une tâche est un cas particulier d'un processus TurboIntegrator parent qui appelle un processus enfant TurboIntegrator planifié pour s'exécuter à une heure spécifique. Vous pouvez utiliser deux types de tâches.

### A validation unique

Dans l'arborescence de portée/d'exécution d'une tâche à validation unique, un sous-ensemble temporaire portant le même nom ne peut être créé que pour un seul processus TurboIntegrator enfant.

### A validation multiple

Dans l'arborescence de portée/d'exécution d'une tâche à validation multiple, qui valide après chaque processus TurboIntegrator enfant, chaque processus TurboIntegrator enfant peut créer un sous-ensemble temporaire portant le même nom car un sous-ensemble temporaire n'est pas conservé après une validation.

Pendant la durée de son existence, un sous-ensemble temporaire a priorité sur un sous-ensemble public ou privé portant le même nom. Si une autre fonction TurboIntegrator fait référence à un sous-ensemble qui existe à la fois à l'état temporaire et à l'état permanent, la fonction s'applique au sous-ensemble temporaire.

Aucun verrouillage n'est associé à un sous-ensemble temporaire car ce type de sous-ensemble n'est jamais enregistré. Cette particularité peut améliorer les performances car TurboIntegrator n'a pas à attendre la libération des verrous pour utiliser un sous-ensemble temporaire.

## Syntaxe

```
SubsetCreate(DimName, SubName, [AsTemporary]);
```

Argument	Description
DimName	Dimension parent du sous-ensemble que vous créez.
SubName	Nom à affecter au sous-ensemble.

Argument	Description
AsTemporary	Argument facultatif qui indique si le sous-ensemble créé est temporaire. 1 indique un sous-ensemble temporaire. 0 indique un sous-ensemble permanent.  Si cet argument est omis, le sous-ensemble est permanent.

### Exemple

```
SubsetCreate('Region', 'Northern Europe', 1);
```

Cet exemple crée le sous-ensemble temporaire Northern Europe pour la dimension Region. Vous pouvez utiliser SubsetElementInsert pour ajouter des éléments au sous-ensemble.

## SubsetCreateByMDX

SubsetCreateByMDX crée un sous-ensemble public basé sur une expression MDX transmise.

Lorsque l'argument AsTemporary a pour valeur 1, le sous-ensemble est temporaire et est conservé pendant la durée du processus TurboIntegrator ou de la tâche à validation unique qui le crée. Si un processus TurboIntegrator parent appelle des processus TurboIntegrator enfant à l'aide de la fonction ExecuteProcess ou ExecuteProcessWithReturn et que le sous-ensemble temporaire est créé dans l'un de ces processus TurboIntegrator enfant, le sous-ensemble est conservé pendant la durée du processus TurboIntegrator parent.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Remarque :

Dans TM1 Server version 11.2.0 et précédente, les vues et les sous-ensembles temporaires étaient visibles et pouvaient être utilisés uniquement par le processus qui les avait créés et par l'un de ses processus enfant. Les vues et les sous-ensembles temporaires n'étaient pas visibles pour les processus ancêtre et les processus apparentés. Vous pouviez créer des sous-ensembles portant un nom identique dans des processus enfant apparentés dotés du même processus parent.

Dans TM1 Server version 11.3.0 et suivante, ces sous-ensembles temporaires sont visibles pour les processus ancêtre et les processus apparentés. Si un processus TurboIntegrator parent "A" appelle deux processus TurboIntegrator enfant "A1" et "A2" et que le processus TurboIntegrator "A1" crée un sous-ensemble temporaire "S", le sous-ensemble temporaire "S" existe pendant la durée du processus TurboIntegrator parent "A". Vous ne pouvez pas créer un sous-ensemble temporaire portant le nom "S" dans le processus TurboIntegrator apparenté "A2" car le sous-ensemble est visible et utilisable par les processus apparentés "A1" et "A2".

Une tâche représente un cas spécial où un processus TurboIntegrator parent appelle un processus TurboIntegrator enfant. Vous pouvez utiliser deux types de tâches.

#### A validation unique

Dans l'arborescence de portée/d'exécution d'une tâche à validation unique, un sous-ensemble temporaire portant le même nom ne peut être créé que pour un seul processus TurboIntegrator enfant.

#### A validation multiple

Dans l'arborescence de portée/d'exécution d'une tâche à validation multiple, qui valide après chaque processus TurboIntegrator enfant, chaque processus TurboIntegrator enfant peut créer un sous-ensemble temporaire portant le même nom car un sous-ensemble temporaire n'est pas conservé après une validation.

Pendant la durée de son existence, un sous-ensemble temporaire a priorité sur un sous-ensemble public ou privé portant le même nom. Si une autre fonction TurboIntegrator fait référence à un sous-ensemble

qui existe à la fois à l'état temporaire et à l'état permanent, la fonction s'applique au sous-ensemble temporaire.

Aucun verrouillage n'est associé à un sous-ensemble temporaire car ce type de sous-ensemble n'est jamais enregistré. Cette particularité peut améliorer les performances car TurboIntegrator n'a pas à attendre la libération des verrous pour utiliser un sous-ensemble temporaire.

## Syntaxe

```
SubsetCreatebyMDX(SubName, MDX_Expression, [AsTemporary]);
```

Argument	Description
SubName	Nom à affecter au sous-ensemble.
MDX_Expression	Expression MDX renvoyant un sous-ensemble
AsTemporary	Argument facultatif qui indique si le sous-ensemble créé est temporaire. 1 indique un sous-ensemble temporaire. 0 indique un sous-ensemble permanent.  Si cet argument est omis, le sous-ensemble est permanent.

## Exemple

```
SubsetCreatebyMDX('0-level months', '{TM1SORT( {TM1FILTERBYLEVEL( {TM1SUBSETALL([month] )}, 0)}, ASC)}', 1);
```

Cet exemple crée un sous-ensemble temporaire nommé '0-level months' basé sur une expression MDX qui renvoie un sous-ensemble composé de tous les éléments de niveau 0 dans la dimension Month, triés par ordre alphabétique croissant.

## Exemple de sous-ensemble temporaire utilisé par le processus parent

Dans cet exemple, deux processus sont créés. Le processus parent, Process-A, appelle le processus enfant, Process-B. Process-B crée un sous-ensemble temporaire que Process-A utilise comme source de données.

Process-A/prolog:

```
ExecuteProcess('Process-B');  
DatasourceDimensionSubset = 'My2003Months';
```

Process-B/prolog:

```
SubsetCreateByMdx('My2003Months', '{[plan_time].[Jan-2003],[plan_time].[Jul-2003]}', 1);
```

## SubsetDeleteAllElements

SubsetDeleteAllElements supprime tous les éléments d'un sous-ensemble public.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SubsetDeleteAllElements(DimName, SubsetName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez supprimer des éléments.
SubsetName	Sous-ensemble dont vous voulez supprimer des éléments. Ce sous-ensemble doit être public. TurboIntegrator n'a pas accès aux objets privés.

### Exemple

```
SubsetDeleteAllElements('Région', 'Europe centrale');
```

Cet exemple supprime tous les éléments du sous-ensemble Central Europe de la dimension Region.

## SubsetDestroy

SubsetDestroy supprime un sous-ensemble de la base de données TM1 .

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetDestroy(DimName, SubName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble que vous supprimez.
SubName	Nom du sous-ensemble à supprimer.

### Exemple

```
SubsetDestroy('Région', 'Europe septentrionale');
```

Cet exemple supprime le sous-ensemble Northern Europe de la dimension Region.

## SubsetElementDelete

SubsetElementDelete supprime un élément d'un sous-ensemble.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetElementDelete(DimName, SubName, Index);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez supprimer un élément.
SubName	Sous-ensemble dont vous voulez supprimer un élément.
Index	Numéro d'index de l'élément à supprimer du sous-ensemble.

## Exemple

```
SubsetElementInsert('Région', 'Europe septentrionale', 'Finlande' 3);
```

Cet exemple supprime le troisième élément du sous-ensemble Northern Europe de la dimension Region.

## SubsetElementExists

SubsetElementExists détermine si un élément spécifique existe dans un sous-ensemble public spécifique sur le serveur à partir duquel un processus TurboIntegrator est exécuté. SubsetElementExists ne permet pas de déterminer si un élément existe dans un sous-ensemble privé.

Si l'élément existe dans le sous-ensemble spécifié, la fonction renvoie 1 ; sinon, elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SubsetElementExists(DimName, SubsetName, ElementName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble contenant l'élément dont vous souhaitez confirmer l'existence.
SubsetName	Sous-ensemble public contenant l'élément dont vous souhaitez confirmer l'existence.
ElementName	Élément dont vous souhaitez confirmer l'existence.

## Exemple

```
SubsetElementExists('Region', 'Europe', 'Italy');
```

Cet exemple détermine si l'élément Italy existe dans le sous-ensemble Europe de la dimension Region.

## SubsetElementGetIndex

SubsetElementGetIndex extrait l'index d'un élément d'un sous-ensemble. La fonction renvoie l'index de la première occurrence de l'élément spécifié.

Si l'élément n'existe pas dans le sous-ensemble ou qu'il est introuvable, la valeur zéro est renvoyée. Si la dimension ou le sous-ensemble est introuvable ou qu'un index de début hors plage est spécifié, une erreur est générée et la fonction TurboIntegrator est arrêtée.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SubsetElementGetIndex(DimName, SubsetName, ElementName, StartIndex);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
SubsetName	Sous-ensemble qui contient l'élément.
ElementName	Nom d'élément (ou alias) à rechercher dans le sous-ensemble.

Argument	Description
StartIndex	Numéro d'index à partir duquel la recherche doit commencer. La valeur doit être comprise entre 1 et la taille du sous-ensemble.

### Exemple

```
SubsetElementGetIndex('Region', 'Europe', 'Italy', 3);
```

Cet exemple extrait l'index d'Italy du sous-ensemble Europe de la dimension Region. La recherche commence à l'index 3.

## SubsetElementInsert

SubsetElementInsert ajoute un élément à un sous-ensemble existant.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetElementInsert(DimName, SubName, ElName, Position);
```

Argument	Description
DimName	Dimension parent du sous-ensemble auquel vous voulez ajouter un élément.
SubName	Nom du sous-ensemble auquel vous ajoutez un élément.
ElName	Nom de l'élément à ajouter au sous-ensemble. Cet élément doit exister dans la base de données TM1.
Position	Valeur indiquant la position d'index de l'élément dans le sous-ensemble.

### Exemple

```
SubsetElementInsert('Région', 'Europe septentrionale', 'Finlande' 3);
```

Cet exemple ajoute l'élément Finland au sous-ensemble Northern Europe de la dimension Region. Finland est le troisième élément dans la définition de sous-ensemble.

## SubsetExists

SubsetExists détermine si un sous-ensemble public spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté.

Cette fonction renvoie 1 si le sous-ensemble existe sur le serveur, sinon, elle renvoie 0. Notez que cette fonction ne peut pas être utilisée pour déterminer l'existence de sous-ensembles privés.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetExists(DimName, SubsetName);
```

Argument	Description
DimName	Nom de la dimension parent du sous-ensemble dont vous voulez confirmer existence
SubsetName	Nom du sous-ensemble public dont vous voulez confirmer l'existence.

### Exemple

```
SubsetExists('Région', 'Europe septentrionale');
```

Cet exemple détermine si le sous-ensemble Northern Europe du cube Region existe sur le serveur.

### SubsetExpandAboveSet

SubsetExpandAboveSet définit la propriété Développer au-dessus pour un sous-ensemble. La fonction renvoie 1 en cas de succès, 0 en cas d'échec.

Quand cette propriété a la valeur TRUE, les enfants d'une consolidation sont affichés au-dessus de la consolidation quand celle-ci apparaît sur une ligne, et à gauche de la consolidation quand celle-ci apparaît dans une colonne.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetExpandAboveSet( DimName, SubsetName, ExpandAboveFlag);
```

Argument	Description
DimName	Dimension parent du sous-ensemble pour laquelle vous voulez définir la propriété "Développer au-dessus".
SubsetName	Sous-ensemble pour lequel vous voulez définir la propriété "Développer au-dessus".
ExpandAboveFlag	Réglez ExpandAboveFlag à 1 pour définir la propriété "Développer au-dessus" à TRUE. Quand cette propriété a la valeur TRUE, les consolidations sont développées au-dessus sur les lignes et à gauche sur les colonnes.  Réglez ExpandAboveFlag à 0 pour définir la propriété "Développer au-dessus" à FALSE. Quand cette propriété a la valeur FALSE, les consolidations sont développées au-dessous sur les lignes et à droite sur les colonnes.

### Exemple

```
SubsetExpandAboveSet('Région', 'Europe', 1 );
```

Cet exemple définit la propriété "Développer au-dessus" à TRUE pour le sous-ensemble Europe de la dimension Region.



## SubsetFormatStyleSet

SubsetFormatStyleSet applique un style d'affichage existant à un sous-ensemble nommé.

Les styles d'affichage sont définis pour des éléments spécifiques. Si vous appliquez un style d'affichage existant à un sous-ensemble comprenant des éléments qui ne sont pas inclus dans le style d'affichage, aucun formatage n'est appliqué à ces éléments.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetFormatStyleSet( DimName, SubsetName, FormatName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble auquel vous voulez ajouter un style d'affichage.
SubsetName	Nom du sous-ensemble auquel vous appliquez un style d'affichage.
FormatName	Nom du style d'affichage existant à appliquer au sous-ensemble.

### Exemple

```
SubsetFormatStyleSet ('Région', 'Europe septentrionale', 'GrasMonnaieJustificationgauche');
```

Cet exemple applique le style d'affichage BoldCurrencyLeftJustified au sous-ensemble Northern Europe de la dimension Region.

## SubsetGetElementName

SubsetGetElementName renvoie le nom de l'élément à un endroit spécifié de l'index dans un sous-ensemble donné.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetGetElementName(DimName, SubsetName, ElementIndex);
```

Argument	Description
DimName	Parent du sous-ensemble depuis lequel vous voulez récupérer un nom d'élément.
SubsetName	Sous-ensemble depuis lequel vous voulez récupérer un nom d'élément.
ElementIndex	Nombre représentant la position dans le sous-ensemble de l'élément à récupérer.

### Exemple

```
SubsetGetElementName('Region', 'Americas', 4);
```

Cet exemple renvoie le quatrième élément du sous-ensemble Amérique de la dimension Region.

## SubsetGetSize

Cette fonction renvoie le nombre d'éléments d'un sous-ensemble.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetGetSize(DimName, SubsetName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez déterminer la taille.
SubsetName	Sous-ensemble dont vous voulez déterminer la taille.

### Exemple

```
SubsetGetSize('Region', 'EurAsia');
```

Cette fonction renvoie le nombre d'éléments du sous-ensemble EurAsia de la dimension Region.

## SubsetIsAllSet

SubsetIsAllSet définit un sous-ensemble pour qu'il utilise tous les éléments de la dimension parent. SubsetIsAllSet renvoie 1 si la définition réussit, sinon elle renvoie 0.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
SubsetIsAllSet( DimName, SubName, Flag );
```

Argument	Description
DimName	Dimension parent du sous-ensemble dont vous voulez utiliser tous les éléments.
SubName	Sous-ensemble dont vous voulez utiliser tous les éléments de dimension.
Flag	Une valeur non nulle indique que le sous-ensemble utilise tous les éléments en cours de la dimension parent et est mis à jour dynamiquement pour utiliser tous les éléments de la dimension parent chaque fois que le sous-ensemble est appelé.  La spécification d'une valeur zéro fige les éléments du sous ensemble comme étant l'ensemble en cours de tous les éléments de la dimension parent. A l'avenir, le sous-ensemble ne sera pas mis à jour dynamiquement pour utiliser tous les éléments de la dimension.

## SubsetMDXGet

SubsetMDXGet extrait l'expression MDX utilisée pour créer un sous-ensemble.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SubsetMDXGet(DimName, SubName);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
SubName	Sous-ensemble pour lequel vous voulez extraire l'expression MDX.

## Exemple

```
mdxString = SubsetMDXGet('Cities', 'testsubset');
```

## SubsetMDXSet

SubsetMDXSet retire l'expression MDX d'un sous-ensemble dynamique et convertit le sous-ensemble dynamique en sous-ensemble statique sans verrouiller la dimension sous-jacente. Le contenu du sous-ensemble est conservé mais l'expression est supprimée. Vous pouvez également remplir le sous-ensemble à l'aide d'une nouvelle expression MDX.

Tous les éléments qui existaient au moment de l'appel sont enregistrés en tant qu'éléments du nouveau sous-ensemble statique. Cette fonction renvoie le nombre d'éléments contenus dans le nouveau sous-ensemble.

**Remarque :** Cette fonction n'est exploitable qu'avec les sous-ensembles publics.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
SubsetMDXSet(DimName, SubName, MDX_expression);
```

Argument	Description
DimName	Dimension parent du sous-ensemble.
SubName	Sous-ensemble pour lequel vous voulez retirer l'expression MDX.
MDX_expression	Expression MDX que vous souhaitez insérer dans le sous-ensemble. Tous les éléments sont supprimés et le sous-ensemble est rempli par les éléments définis par la nouvelle expression MDX. Si l'expression MDX est invalide, le traitement TurboIntegrator s'arrête, le contenu du sous-ensemble n'est pas modifié et une erreur est enregistrée.  Si vous n'utilisez pas ce paramètre, entrez une chaîne vide, comme illustré dans l'exemple.

## Exemple

```
countMem = SubsetMDXSet('Cities', 'testsubset', '');
```

## Fonctions TurboIntegrator de manipulation de vues

Ces fonctions servent pour la manipulation des vues.

## PublishView

PublishView publie une vue privée nommée sur le serveur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
PublishView(Cube, View, PublishPrivateSubsets, OverwriteExistingView);
```

Argument	Description
Cube	Nom du cube contenant la vue privée à publier.
Vue	Nom de la vue privée à publier.
PublishPrivateSubsets	<p>Cet argument booléen (1 ou 0) détermine si des sous-ensembles privés présents dans la vue doivent aussi être publiés.</p> <p>Si PublierSousEnsemblesPrivés est vrai (1), tous les sous-ensembles privés utilisés dans la vue sont publiés en même temps que la vue.</p> <p>Si cet argument est faux (0), les sous-ensembles privés ne sont pas publiés. Une vue publique ne peut pas contenir de sous-ensembles privés et, par conséquent, la vue n'est pas publiée et une erreur est entrée dans le fichier journal de TurboIntegrator.</p> <p><b>Remarque :</b> Si un sous-ensemble privé comporte un autre sous-ensemble privé sous forme de consolidation définie par l'utilisateur, le sous-ensemble ne peut jamais être publié en utilisant la fonction PublishView, quelle que soit la valeur de l'argument PublierSousEnsemblesPrivés.</p>
OverwriteExistingView	<p>Cet argument booléen (1 or 0) détermine si, le cas échéant, une vue publique existante ayant un nom identique doit être remplacée quand la vue privée est publiée.</p> <p>Si RemplacerVueExistante est vrai (1), toute vue publique existante ayant un nom identique est remplacée quand la vue privée est publiée.</p> <p>Si l'argument est faux (0), la vue publique n'est pas remplacée, la vue privée n'est pas publiée et une erreur est entrée dans le fichier journal de TurboIntegrator.</p>

## DisableMTQViewConstruct

DisableMTQViewConstruct désactive le traitement de requêtes à unités d'exécution multiples lors du calcul d'une vue à utiliser comme source de données TurboIntegrator pour un processus TurboIntegrator unique. Lorsque MTQQuery=T est défini dans le fichier tms1.cfg, la fonction DisableMTQViewConstruct peut être appelée pour remplacer cette valeur pour un processus TurboIntegrator.

Cette fonction doit apparaître dans le prologue ; elle n'a pas d'effet sur une autre procédure au sein d'un processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

**Remarque :** Si la valeur du paramètre **MTQ** est 1 (ou OFF), cette fonctionnalité est entièrement désactivée et ne peut pas être remplacée.

La valeur de **MTQQuery** peut être remplacée pour un processus TurboIntegrator unique via l'appel de la fonction `DisableMTQViewConstruct`.

Si **MTQQuery=T** est indiqué (valeur par défaut), la fonction `DisableMTQViewConstruct` peut être appelée pour désactiver la fonctionnalité pour des processus TurboIntegrator spécifiques.

Une fois que la fonction `EnableMTQViewConstruct` est activée dans un processus, elle ne peut être désactivée que sur la dernière ligne de la section Epilogue. Si vous tentez d'utiliser `DisableMTQViewConstruct` ailleurs dans le processus, le processus ne sera pas compilé.

Si le mode est activé dans un processus TurboIntegrator, il reste activé tant qu'il n'est pas explicitement désactivé ou tant que la tâche n'est pas terminée. Autrement dit, vous pouvez activer le mode dans un processus, puis exécuter une série de processus TurboIntegrator avant de le désactiver.

### Exemple

Utilisez les commandes TurboIntegrator suivantes pour désactiver le traitement des requêtes à unités d'exécution multiples lors du calcul d'une vue à utiliser comme source de données TurboIntegrator pour un processus TurboIntegrator unique.

```
DisableMTQViewConstruct()
```

Voir aussi [«EnableMTQViewConstruct»](#), à la page 397.

## EnableMTQViewConstruct

`EnableMTQViewConstruct` active le traitement de requêtes à unités d'exécution multiples lors du calcul d'une vue à utiliser comme source de données TurboIntegrator pour un processus TurboIntegrator unique. Lorsque **MTQQuery=F** est défini dans le fichier `tms1.cfg`, la fonction `EnableMTQViewConstruct` peut être appelée pour remplacer cette valeur pour un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

**Remarque :** Si la valeur du paramètre **MTQ** est 1 (ou OFF), cette fonctionnalité est entièrement désactivée et ne peut pas être remplacée.

La valeur de **MTQQuery** peut être remplacée pour un processus TurboIntegrator unique via l'appel de la fonction `EnableMTQViewConstruct`.

Si **MTQQuery=F** est défini dans le fichier `tms1.cfg` sur le serveur où cette fonction est exécutée, la fonction `EnableMTQViewConstruct` peut être appelée pour remplacer cette valeur pour un processus TurboIntegrator unique.

Vous pouvez activer `EnableMTQViewConstruct` dans la section Prologue ou Epilogue d'un processus TurboIntegrator. Par souci d'efficacité, activez `EnableMTQViewConstruct` dans la première instruction (ou du moins le plus près possible) de la section Prologue de votre processus.

### Exemple

Utilisez les commandes TurboIntegrator suivantes pour activer le traitement de requêtes à unités d'exécution multiples lors du calcul d'une vue à utiliser comme source de données TurboIntegrator pour un processus TurboIntegrator unique.

```
EnableMTQViewConstruct()
```

Voir aussi «DisableMTQViewConstruct», à la page 396.

Une fois que la fonction EnableMTQViewConstruct est activée dans un processus, elle ne peut être désactivée que sur la dernière ligne de la section Epilogue. Si vous tentez de désactiver DisableMTQViewConstruct ailleurs dans le processus, le processus ne sera pas compilé.

Si le mode est activé dans un processus TurboIntegrator, il reste activé tant qu'il n'est pas explicitement désactivé ou tant que la tâche n'est pas terminée. Autrement dit, vous pouvez activer le mode dans un processus, puis exécuter une série de processus TurboIntegrator avant de le désactiver.

## ViewColumnDimensionSet

ViewColumnDimensionSet définit une dimension de colonne pour une vue TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewColumnDimensionSet(CubeName, ViewName, DimName, StackPosition);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous définissez la dimension en colonne.
ViewName	Vue pour laquelle vous définissez la dimension en colonne.
DimName	Dimension à définir comme dimension en colonne pour la vue.
StackPosition	Nombre indiquant la position de pile de la dimension dans la vue. Ce nombre est en base 1. 1 indique la position la plus en haut de la pile. 2 indique une position sous 1, etc.

### Exemple

```
ViewColumnDimensionSet('Ventres98', 'Trim1', 'Mois', 1);
```

Cet exemple définit Month comme dimension en colonne pour la vue 1Quarter du cube 98sales. S'il s'agit de dimensions en colonne empilées, la dimension Month est placée le plus en haut.

## ViewColumnSuppressZeroesSet

ViewColumnSuppressZeroesSet désactive ou active l'affichage des colonnes contenant uniquement des valeurs zéro dans une vue de cube TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewColumnSuppressZeroesSet(Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous voulez supprimer ou activer l'affichage des valeurs zéro.

Argument	Description
ViewName	Vue pour laquelle vous voulez activer ou supprimer l'affichage des zéros.
Flag	Valeur binaire qui active ou supprime les zéros. Spécifiez 1 pour supprimer l'affichage des colonnes contenant seulement des zéros dans la vue. Spécifiez 0 pour activer l'affichage des colonnes contenant seulement des zéros.

### Exemple

```
ViewColumnSuppressZeroesSet('99sales', '1st QuarterActuals', 1);
```

Cet exemple supprime l'affichage des colonnes renfermant uniquement des zéros dans la vue 1st Quarter Actuals du cube 99sales.

## ViewConstruct

La fonction ViewConstruct construit, pré-calcule et stocke une vue Stargate dans la mémoire d'un serveur. Elle est utile pour le pré-calcul et le stockage de vues volumineuses et permet d'y accéder rapidement après le chargement ou la mise à jour de données.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewConstruct(CubeName, ViewName);
```

Argument	Description
CubeName	Cube à partir duquel vous voulez construire la vue.
ViewName	Vue à construire. Cette vue doit être une vue publique existante et sur le serveur.

### Exemple

```
Viewconstruct('99sales', '1st Quarter Actuals');
```

Cet exemple supprime la vue 1st Quarter Actuals du cube 99sales.

## ViewCreate

ViewCreate crée une vue vide d'un cube spécifié.

Lorsque l'argument facultatif AsTemporary est défini sur 1, la vue est temporaire et n'est conservée que pendant la durée du processus ou de la tâche TurboIntegrator qui l'a créée.

### Remarque :

Dans TM1 Server version 11.2.0 et précédente, les vues temporaires étaient visibles et pouvaient être utilisées uniquement par le processus qui les avait créées et par l'un de ses processus enfant. Les vues temporaires n'étaient pas visibles pour les processus ancêtre et les processus apparentés. Vous pouviez créer des vues portant un nom identique dans des processus enfant apparentés dotés du même processus parent.

Dans TM1 Server version 11.3.0 et suivante, ces vues temporaires sont visibles pour les processus ancêtre et les processus apparentés. Si un processus TurboIntegrator parent "A" appelle deux processus TurboIntegrator enfant "A1" et "A2" et que le processus TurboIntegrator "A1" enfant crée une vue temporaire "S", la vue temporaire "S" existe pendant la durée du processus parent TurboIntegrator "A". Vous ne pouvez pas créer une vue temporaire portant le nom "S" dans le processus TurboIntegrator apparenté "A2" car la vue est visible et utilisable par les processus apparentés "A1" et "A2".

Pendant la durée de son existence, une vue temporaire est prioritaire par rapport à une vue publique du même nom. Une autre fonction TurboIntegrator qui référencerait une vue existant à la fois à l'état temporaire et à l'état permanent utiliserait la vue temporaire.

Les objets temporaires possèdent une portée de transaction. Lorsqu'une transaction est validée, tous les objets temporaires sont supprimés. Si une tâche est exécutée dans le mode à validation unique, où tous les processus de la tâche sont exécutés de manière logique dans le contexte d'une transaction unique, les objets temporaires créés dans un processus sont conservés, restent visibles et peuvent être utilisés dans les processus exécutés ultérieurement par la tâche. En revanche, dans le mode à validations multiples, ces processus sont supprimés au moment de la validation de la transaction qui a encapsulé l'exécution du processus ayant créé l'objet temporaire.

Aucun verrouillage n'est associé aux vues temporaires, car elles ne sont jamais enregistrées. Cette particularité peut améliorer les performances, car TurboIntegrator n'a jamais besoin d'attendre la libération des verrous pour utiliser une vue temporaire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewCreate(Cube, ViewName, <AsTemporary>);
```

Argument	Description
Cube	Cube parent de la vue que vous créez.
ViewName	Nom à affecter à la vue.
AsTemporary	Argument facultatif qui indique si la vue créée est temporaire. 1 indique une vue temporaire, 0 indique une vue permanente.  Si cet argument est omis, la vue est permanente.

## Exemple

Cet exemple crée une vue temporaire nommée 1st Quarter Actuals à partir du cube Sales.

```
ViewCreate('Sales', '1st Quarter Actuals', 1);
```

## ViewCreateByMDX

ViewCreateByMDX crée une vue avec une expression MDX spécifiée.

Lorsque l'argument facultatif AsTemporary est défini sur 1, la vue est temporaire et n'est conservée que pendant la durée du processus ou de la tâche TurboIntegrator qui l'a créée.

### Remarque :

Dans TM1 Server version 11.2.0 et précédente, les vues temporaires étaient visibles et pouvaient être utilisées uniquement par le processus qui les avait créées et par l'un de ses processus enfant. Les vues temporaires n'étaient pas visibles pour les processus ancêtre et les processus apparentés. Vous pouviez créer des vues portant un nom identique dans des processus enfant apparentés dotés du même processus parent.



Dans TM1 Server version 11.3.0 et suivante, ces vues temporaires sont visibles pour les processus ancêtre et les processus apparentés. Si un processus TurboIntegrator parent "A" appelle deux processus TurboIntegrator enfant "A1" et "A2" et que le processus TurboIntegrator "A1" enfant crée une vue temporaire "S", la vue temporaire "S" existe pendant la durée du processus parent TurboIntegrator "A". Vous ne pouvez pas créer une vue temporaire portant le nom "S" dans le processus TurboIntegrator apparenté "A2" car la vue est visible et peut être utilisée par les processus apparentés "A1" et "A2".

Pendant la durée de son existence, une vue temporaire est prioritaire par rapport à une vue publique du même nom. Une autre fonction TurboIntegrator qui référencerait une vue existant à la fois à l'état temporaire et à l'état permanent utiliserait la vue temporaire.

Les objets temporaires possèdent une portée de transaction. Lorsqu'une transaction est validée, tous les objets temporaires sont supprimés. Si une tâche est exécutée dans le mode à validation unique où tous les processus de la tâche sont exécutés de manière logique dans le contexte d'une transaction unique, les objets temporaires créés dans un processus sont conservés, restent visibles et peuvent être utilisés dans les processus exécutés ultérieurement par la tâche. En revanche, dans le mode à validations multiples, ces processus sont supprimés au moment de la validation de la transaction qui a encapsulé l'exécution du processus ayant créé l'objet temporaire.

Aucun verrouillage n'est associé aux vues temporaires, car elles ne sont jamais enregistrées. Cela peut améliorer les performances car TurboIntegrator n'a pas attendre le déverrouillage pour traiter une vue temporaire.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewCreateByMDX(Cube, ViewName, MDX_expression , <AsTemporary>);
```

Argument	Description
Cube	Cube parent de la vue que vous créez.
ViewName	Nom à affecter à la vue.
MDX_expression	Valeur de chaîne contenant une expression de vue MDX valide.
AsTemporary	Argument facultatif qui indique si la vue créée est temporaire. 1 indique une vue temporaire ; 0 indique une vue permanente.  Si cet argument est omis, la vue est permanente.

## Exemple

Cet exemple, basé sur la base de données Planning Sample, crée une vue temporaire appelée Account dans le cube plan\_BudgetPlan.

```
ViewCreateByMDX('plan_BudgetPlan', 'Account',
  'select {[plan_version].[FY 2003 Budget]} on 0,
  {[plan_business_unit].[10300]} on 1 from plan_budgetplan where
  [plan_department].[200][plan_chart_of_accounts].[41101][plan_exchange_rates].[local]
  [plan_source].[goal][plan_time].[Jan-2003]'
  ,1);
```

## ViewDestroy

ViewDestroy supprime une vue de la base de données TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewDestroy(Cube, ViewName);
```

Argument	Description
Cube	Cube parent de la vue que vous supprimez.
ViewName	Nom de la vue à supprimer.

## Exemple

```
ViewZeroOut('Ventés99', 'Réal Trim 1');
```

Cet exemple supprime la vue 1st Quarter Actuals du cube 99sales.

## ViewExists

ViewExists détermine si une vue publique spécifique existe sur le serveur à partir duquel un processus TurboIntegrator est exécuté. Cette fonction renvoie 1 si la vue existe sur le serveur, sinon elle renvoie 0. Notez que cette fonction ne peut pas être utilisée pour déterminer l'existence de vues privées.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewExists(CubeName, ViewName);
```

Argument	Description
CubeName	Nom du cube parent de la vue dont vous voulez confirmer l'existence.
ViewName	Nom de la vue publique dont vous voulez confirmer l'existence.

## Exemple

```
ViewExists('Inventaire', 'FinFévrier');
```

Cet exemple détermine si la vue Febclosing du cube Inventory existe sur le serveur.

## ViewExtractFilterByTitlesSet

ViewExtractFilterByTitlesSet définit une option permettant d'effectuer des filtrage par titre sur des valeurs consolidées qui sont exclues d'une vue ou des extractions de vue associées.

TM1 autorise le stockage des chaînes sur les valeurs calculées. Lorsque vous excluez une valeur calculée d'une vue ou d'une extraction de vue, vous souhaitez peut-être exclure également la chaîne du message de la vue.

**Remarque :** Cette fonction affecte des vues telles qu'elles existent sur le serveur. La portée de cette fonction ne se limite pas aux extractions générées à partir d'une vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewExtractFilterByTitlesSet (Cube, ViewName, FilterByTitles, Temporary);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous définissez l'option.
ViewName	Vue pour laquelle vous définissez l'option.
FilterByTitles	Valeur binaire qui active ou désactive l'option. Indiquez 0 pour inclure les titres stockés sur des valeurs consolidées. Il s'agit du comportement en cours par défaut. Indiquez 1 pour exclure les titres stockés sur des valeurs consolidées.
Temporary	Valeur booléenne indiquant si les paramètres sont temporaires.

### Exemple

```
ViewExtractFilterByTitlesSet ('99sales', '1st QuarterActuals', 1, 0);
```

### ViewExtractSkipCalcsSet

ViewExtractSkipCalcsSet définit une option permettant d'inclure/exclure des valeurs consolidées dans une vue **et** dans les extractions de vue associées. Une extraction de vue est une vue TM1 exportée comme fichier ASCII (.cma) à virgule de séparation.

**Remarque :** Cette fonction affecte des vues telles qu'elles existent sur le serveur. La portée de cette fonction ne se limite pas aux extractions générées à partir d'une vue.

ViewExtractSkipCalcsSet est l'équivalent de l'option Ignorer les valeurs calculées dans la boîte de dialogue Extraction de vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewExtractSkipCalcsSet (Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous définissez l'option.
ViewName	Vue pour laquelle vous définissez l'option.
Flag	Valeur binaire qui active ou désactive l'option. Spécifiez 1 pour exclure les valeurs calculées de l'extraction de vue. Spécifiez 0 pour inclure les valeurs calculées.

### Exemple

```
ViewExtractSkipCalcsSet ('Ventés99', 'Réel Trim1', 1);
```

Cet exemple active l'option Ignorer les valeurs calculées pour la vue 1st Quarter Actuals. L'extraction de vue ne contiendra aucune valeur calculée.

## Remarque sur les conséquences de l'activation d'une combinaison spécifique de fonctions de manipulation de vues

Supposons que les conditions suivantes soient vraies :

- La mesure est une chaîne
- `ViewExtractSkipCalcsSet = 1`
- `ViewExtractSkipConsolidatedStringsSet = 0` (la fonction n'est pas utilisée)
- `ViewExtractSkipRuleValuesSet = 0` (la fonction n'est pas utilisée)

Dans ce scénario, la sortie diffère selon que la fonction `ViewExtractSkipZeroesSet` est activée ou non.

- Si vous définissez `ViewExtractSkipZeroesSet = 0`, le serveur TM1 énumère toutes les cellules possibles du cube, et pas seulement les cellules de données existantes. Cette situation est plutôt inhabituelle, car l'énumération de toutes les cellules possibles signifie que le nombre de cellules analysées est le produit des tailles de toutes les dimensions du cube. Ce produit peut devenir très grand. Dans ce mode, la fonction `ViewExtractSkipCalcsSet` ignore toutes les cellules consolidées, même si la mesure est une chaîne.
- Si vous définissez `ViewExtractSkipZeroesSet = 1`, le serveur TM1 n'analyse que les cellules réellement dans le cube. Dans ce mode, une chaîne stockée dans une cellule consolidée est considérée comme une simple feuille (après tout, la cellule contient une valeur simple et elle est une feuille). Ainsi, même si la fonction `ViewExtractSkipCalcsSet` est activée, l'entrée n'est pas ignorée car la cellule n'est pas une cellule consolidée calculée. Dans ce cas, si vous ne voulez pas que les entrées soient ignorées, vous devez activer la fonction `ViewExtractSkipConsolidatedStringsSet`.

## ViewExtractSkipConsolidatedStringsSet

`ViewExtractSkipConsolidatedStringsSet` définit une option permettant d'exclure des chaînes sur des valeurs consolidées d'une vue ou des extractions de vue associées. Une extraction de vue est une vue TM1 exportée comme fichier ASCII (.cma) à virgule de séparation.

TM1 autorise le stockage des chaînes sur les valeurs calculées. Lorsque vous excluez une valeur calculée d'une vue ou d'une extraction de vue, vous souhaitez peut-être exclure également la chaîne du message de la vue.

**Remarque :** Cette fonction affecte des vues telles qu'elles existent sur le serveur. La portée de cette fonction ne se limite pas aux extractions générées à partir d'une vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewExtractSkipConsolidatedStringSet (Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous définissez l'option.
ViewName	Vue pour laquelle vous définissez l'option.
Flag	Valeur binaire qui active ou désactive l'option. Spécifiez 0 pour inclure les chaînes stockées sur les valeurs consolidées. Il s'agit du comportement en cours par défaut. Spécifiez 1 pour exclure les chaînes stockées sur les valeurs consolidées.

**Remarque :** Lisez la Remarque sur les conséquences de l'activation d'une combinaison spécifique de fonctions de manipulation de vue.

## Exemple

```
ViewExtractSkipConsolidatedStringsSet ('99sales', '1st QuarterActuals', 1);
```

Cet exemple active l'option Ignorer la règle pour la chaîne consolidée pour l'extraction créée à partir de la vue 1st QuarterActuals. L'extraction ne contiendra aucune chaîne sur la valeur consolidée.

## ViewExtractSkipRuleValuesSet

ViewExtractSkipRuleValuesSet définit une option permettant d'inclure/exclure des valeurs calculées par règle dans une vue **et** les extractions de vue associées. Une extraction de vue est une vue TM1 exportée comme fichier ASCII (.cma) à virgule de séparation.

ViewExtractSkipRuleValuesSet est l'équivalent de l'option Ignorer les valeurs calculées par règle dans la boîte de dialogue Extraction de vue.

**Remarque :** Cette fonction affecte des vues telles qu'elles existent sur le serveur. La portée de cette fonction ne se limite pas aux extractions générées à partir d'une vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewExtractSkipRuleValuesSet (Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous définissez l'option.
ViewName	Vue pour laquelle vous définissez l'option.
Flag	Valeur binaire qui active ou désactive l'option. Spécifiez 1 pour exclure de la vue les valeurs calculées par règle. Spécifiez 0 pour inclure les valeurs calculées par règle.

**Remarque :** Lisez la [Remarque sur les conséquences de l'activation d'une combinaison spécifique de fonctions de manipulation de vue](#).

## Exemple

```
ViewExtractSkipRuleValuesSet ('99sales', '1st QuarterActuals', 1);
```

Cet exemple active l'option Ignorer les valeurs calculées par règle pour l'extraction créée à partir de la vue 1st QuarterActuals. L'extraction de vue ne comprendra aucune valeur calculée par règle.

## ViewExtractSkipZeroesSet

ViewExtractSkipZeroesSet définit une option permettant d'inclure/exclure des valeurs nulles dans une vue **et** dans les extractions de vue associées. Une extraction de vue est une vue TM1 exportée comme fichier ASCII (.cma) à virgule de séparation.

ViewExtractSkipZeroesSet est l'équivalent de l'option Ignorer les valeurs zéro/nulles dans la boîte de dialogue Extraction de vue.

**Remarque :** Cette fonction affecte des vues telles qu'elles existent sur le serveur. La portée de cette fonction ne se limite pas aux extractions générées à partir d'une vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewExtractSkipZeroesSet (Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous définissez l'option Ignorer les zéros.
ViewName	La vue pour laquelle vous définissez l'option Ignorer les zéros.
Flag	Valeur binaire qui active ou désactive l'option. Spécifiez 1 pour exclure les zéros de l'extraction. Spécifiez 0 pour inclure les zéros.  Si UNDEFVALS est utilisé pour représenter les zéros, les valeurs ne sont pas exclues lorsque l'argument Flag a pour valeur 1.

**Remarque :** Lisez la Remarque sur les conséquences de l'activation d'une combinaison spécifique de fonctions de manipulation de vue.

## Exemple

```
ViewExtractSkipZeroesSet ('Ventés99', 'Réel Trim1', 1);
```

Cet exemple active l'option Ignorer les valeurs zéro pour l'extraction créée à partir de la vue 1st Quarter Actuals. L'extraction de vue ne comprendra aucune valeur nulle ou zéro.

## ViewMDXSet

ViewMDXSet définit l'expression MDX d'une vue MDX existante.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ViewMDXSet(Cube, ViewName, MDX_expression);
```

Argument	Description
Cube	Cube parent de la vue que vous créez.
ViewName	Nom à affecter à la vue.
MDX_expression	Valeur de chaîne contenant une expression de vue MDX valide.

## Exemple

```
ViewMDXSet('Sales', 'Account',  
  "select {[plan_version].[FY 2003 Budget]} on 0,  
  {[plan_business_unit].[10300]} on 1 from plan_budgetplan where  
  [plan_department].[200][plan_chart_of_accounts].[41101][plan_exchange_rates].[local]  
  [plan_source].[goal][plan_time].[Jan-2003]"  
  );
```

Cet exemple définit l'expression MDX de la vue "Account" du cube "Sales".

## ViewMDXGet

ViewMDXGet extrait l'expression MDX d'une vue MDX existante.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewMDXGet(Cube, ViewName);
```

Argument	Description
Cube	Cube parent de la vue que vous créez.
ViewName	Nom à affecter à la vue.

### Exemple

```
ViewMDXGet('Sales', 'Account');
```

Cet exemple extrait l'expression MDX de la vue "Account".

## ViewRowDimensionSet

ViewRowDimensionSet définit une dimension de ligne pour une vue TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewRowDimensionSet(CubeName, ViewName, DimName, StackPosition);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous définissez la dimension de ligne.
ViewName	Vue pour laquelle vous définissez la dimension de ligne.
DimName	Dimension à définir comme dimension de ligne pour la vue.

Argument	Description
StackPosition	<p>Nombre indiquant la position de pile de la dimension dans la vue. Ce nombre est en base 1. 1 indique la position la plus à gauche de la pile. 2 indique une position à droite de 1, etc.</p> <p><b>Remarque :</b> Le client TM1 peut définir un paramètre Tm1p.ini (BrowseDisplayReadsRightToLeft=T) inversant l'orientation des données dans Cube Viewer. Lorsque l'orientation des données est inversée, la position des piles est également inversée. 1 indique la position la plus à droite de la pile. 2 indique une position à gauche de 1, etc.</p>

### Exemple

```
ViewRowDimensionSet('98sales', 'Quarter1', 'Month',1)
```

Cet exemple définit Month comme dimension de ligne pour la vue 1Quarter du cube 98sales. S'il s'agit de dimensions de ligne empilées, la dimension Month est placée le plus à gauche.

### ViewRowSuppressZeroesSet

ViewRowSuppressZeroesSet désactive ou active l'affichage des lignes contenant uniquement des valeurs zéro dans une vue de cube TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewRowSuppressZeroesSet(Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous voulez supprimer ou activer l'affichage des valeurs zéro.
ViewName	Vue pour laquelle vous voulez activer ou supprimer l'affichage des zéros.
Flag	Valeur binaire qui active ou supprime les zéros. Spécifiez 1 pour supprimer l'affichage des lignes contenant seulement des zéros dans la vue. Spécifiez 0 pour activer l'affichage des lignes contenant seulement des zéros.

### Exemple

```
ViewRowSuppressZeroesSet('Ventés99', 'Réal Trim1', 1);
```

Cet exemple supprime l'affichage des lignes renfermant uniquement des zéros dans la vue 1st Quarter Actuals du cube 99sales.



## ViewSubsetAssign

ViewSubsetAssign affecte un sous-ensemble nommé à une vue de cube.

**Remarque :** Il est possible de créer un sous-ensemble temporaire à l'aide des fonctions [CreateSubset](#) ou [CreateSubsetByMDX](#). Si vous tentez d'utiliser ViewSubsetAssign pour affecter un sous-ensemble temporaire à une vue permanente, la fonction échouera et vous recevrez un message d'erreur.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewSubsetAssign(Cube, ViewName, DimName, SubName);
```

Argument	Description
Cube	Cube parent de la vue à laquelle vous affectez un sous-ensemble.
ViewName	Vue à laquelle vous affectez un sous-ensemble.
DimName	Dimension parent du sous-ensemble que vous affectez à la vue.
SubName	Nom du sous-ensemble à affecter à la vue.

### Exemple

```
ViewSubsetAssign('Ventés99', 'Réal Trim1', 'Mois', 'T1');
```

Cet exemple affecte le sous-ensemble Q1 de la dimension Month à la vue 1st Quarter.

## ViewSuppressZeroesSet

ViewSuppressZeroesSet supprime ou active l'affichage de toutes les lignes et colonnes contenant uniquement la valeur zéro dans une vue de cube TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewSuppressZeroesSet(Cube, ViewName, Flag);
```

Argument	Description
Cube	Cube parent de la vue pour laquelle vous voulez supprimer ou activer l'affichage des valeurs zéro.
ViewName	Vue pour laquelle vous voulez activer ou supprimer l'affichage des zéros.
Flag	Valeur binaire qui active ou supprime les zéros. Spécifiez 1 pour supprimer l'affichage des lignes et des colonnes contenant seulement des zéros dans la vue. Spécifiez 0 pour activer l'affichage des lignes et des colonnes contenant seulement des zéros dans la vue.

## Exemple

```
ViewSuppressZeroesSet('Ventés99', 'Réal Trim1', 1);
```

Cet exemple supprime l'affichage des lignes et colonnes renfermant uniquement des zéros dans la vue 1st Quarter Actuals du cube 99sales.

## ViewTitleDimensionSet

ViewTitleDimensionSet définit une dimension de titre pour une vue TM1.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewTitleDimensionSet(CubeName, ViewName, DimName);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous définissez la dimension en titre.
ViewName	Vue pour laquelle vous définissez la dimension en titre.
DimName	Dimension à définir comme dimension en titre pour la vue.

## Exemple

```
ViewTitleDimensionSet('Ventés98', 'Trim1', 'Mois');
```

Cet exemple définit Month comme dimension de titre pour la vue 1Quarter du cube 98sales.

## ViewTitleElementSet

ViewTitleElementSet définit un élément de titre pour une vue TM1. ViewTitleElementSet est utilisé conjointement avec la fonction ViewTitleDimensionSet.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewTitleElementSet(CubeName, ViewName, DimName, Index);
```

Argument	Description
CubeName	Cube parent de la vue pour laquelle vous définissez l'élément en titre.
ViewName	Vue pour laquelle vous définissez l'élément en titre.
DimName	Dimension parent de l'élément en titre.
Index	Index de la dimension spécifiée indiquant l'élément à définir comme élément en titre.

## Exemple

```
ViewTitleElementSet('Ventés98' , 'Trim1' , 'Modèle', 3);
```

Cet exemple définit le troisième élément de la dimension Model comme élément de titre de la vue Quarter1 du cube 98sales.

## ViewZeroOut

ViewZeroOut affecte la valeur zéro à tous les points de données d'une vue.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
ViewZeroOut(Cube, ViewName);
```

Argument	Description
Cube	Cube parent de la vue que vous voulez définir à zéro.
ViewName	Vue à définir à zéro.

## Exemple

```
ViewZeroOut('Ventés99', 'Réal Trim 1');
```

Cette fonction définit toutes les données individuelles de la vue 1st Quarter Actuals à zéro.

## Fonctions TurboIntegrator diverses

---

Ces fonctions facilitent des tâches diverses.

### AddInfoCubeRestriction

AddInfoCubeRestriction filtre les données InfoCube au fur et à mesure de leur extraction et de leur transfert vers TM1. Utilisez cette fonction pour restreindre le nombre de valeurs à importer pour une caractéristique spécifique. Cette fonction doit être placée dans le prologue. La fonction peut être appelée plusieurs fois pour filtrer plusieurs caractéristiques au cours d'un même processus.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
AddInfoCubeRestriction(STRING CharactName, STRING sign, STRING compOperator,  
STRING lowValue, STRING highValue)
```

Argument	Description
STRING CharactName	Contient le nom technique de la caractéristique à restreindre. Le type de données doit être une chaîne de caractères d'une longueur inférieure ou égale à 30.

Argument	Description
STRING sign	Contient soit I (= inclusif) or E (= exclusif). Exclusif correspond au NON logique pour la restriction définie par cette ligne. Le type de données doit être un caractère de longueur 1.
STRING compOperator	Contient l'opérateur de comparaison. Le type de données doit être une chaîne de caractères de longueur 2. Les opérateurs de comparaison valides sont les suivants : 'EQ' = égal à 'NE' = différent de 'LT' = inférieur à 'GT' = supérieur à 'LE' = inférieur ou égal à 'GE' = supérieur ou égal à 'BT' = compris entre 'NB' = non compris entre
STRING ValeurInf	Contient la valeur inférieure pour l'opérateur spécifiée dans la ligne précédente. Le type de données doit être une chaîne de caractères d'une longueur inférieure ou égale à 60.
STRING ValeurSup	Contient la valeur supérieure pour l'opérateur spécifiée deux lignes plus haut. Le type de données doit être une chaîne de caractères d'une longueur inférieure ou égale à 60. Ces données ne sont nécessaires que pour les opérateurs BT et NB. Dans les autres cas, elles sont ignorées et une chaîne vide doit être utilisée à la place.

### Exemple

L'exemple suivant renvoie toutes les valeurs de caractéristiques comprises entre 1997 et 2000.

```
AddInfoCubeRestriction('0CALYEAR','E','BT','1997','2000');
```

L'exemple suivant renvoie toutes les valeurs de caractéristiques qui ne sont pas comprises entre 1997 et 2000.

```
AddInfoCubeRestriction('0CALYEAR','I','NB','1997','2000');
```

L'exemple suivant renvoie toutes les valeurs de caractéristiques qui ne sont pas égales à USD.

```
AddInfoCubeRestriction('0DOC_CURRCY','I','NE','USD','');
```

### ExecuteJavaN

ExecuteJavaN exécute un processus TurboIntegrator Java™ qui renvoie un nombre. Si vous souhaitez exécuter un processus TurboIntegrator Java qui renvoie une chaîne, utilisez ExecuteJavaS.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ExecuteJavaN('JavaTIClass', ['OptionalParameter1', 'OptionalParameter2', ...] )
```

Argument	Description
JavaTIClass	Nom qualifié complet de la classe TurboIntegrator Java que vous souhaitez exécuter.
OptionalParameters	Paramètres facultatifs qui sont transmis au processus TurboIntegrator Java lui-même. Vous pouvez transmettre autant de paramètres que nécessaire, y compris aucun.  Seules des chaînes peuvent être transmises dans les paramètres, et non des nombres. Vous pouvez utiliser la fonction TurboIntegrator StringToNumber pour transmettre des nombres aux scripts TurboIntegrator Java. Pour plus d'informations, voir "StringToNumber" dans le document <i>TM1 - Guide de référence</i> .

Une classe TurboIntegrator Java, qui renvoie un nombre et peut être appelée par ExecuteJavaN, doit utiliser le modèle suivant :

```
package com.example;  
  
import com.ibm.cognos.tm1.javati.JavaTI;  
  
@JavaTI  
public class MyTestTI {  
    public static double MyTestTI (String [] args) [  
        ...  
        return ...;  
    }  
}
```

## Exemple

```
ExecuteJavaN('com.example.MyTestTI', 'First', 'Second', 'Third');
```

## ExecuteJavaS

ExecuteJavaS exécute un processus TurboIntegrator Java qui renvoie une chaîne. Si vous souhaitez exécuter un processus TurboIntegrator Java qui renvoie un nombre, utilisez ExecuteJavaN.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
ExecuteJavaS('JavaTIClass', ['OptionalParameter1', 'OptionalParameter2', ...] )
```

Argument	Description
JavaTIClass	Nom qualifié complet de la classe TurboIntegrator Java que vous souhaitez exécuter.

Argument	Description
OptionalParameters	<p>Paramètres facultatifs qui sont transmis au processus TurboIntegrator Java lui-même. Vous pouvez transmettre autant de paramètres que nécessaire, y compris aucun.</p> <p>Seules des chaînes peuvent être transmises dans les paramètres, et non des nombres. Vous pouvez utiliser la fonction TurboIntegrator StringToNumber pour transmettre des nombres aux scripts TurboIntegrator Java. Pour plus d'informations, voir "StringToNumber" dans le document <i>TM1 - Guide de référence</i>.</p>

Une classe TurboIntegrator Java, qui renvoie une chaîne et est appelée par ExecuteJavaS, doit utiliser le modèle suivant.

```

package com.example;

import com.ibm.cognos.tm1.javati.JavaTI;

@JavaTI
public class MyTestTI {
    public static String MyTestTI (String [] args) [
        ...
        return ...;
    ]
}

```

### Exemple

```
ExecuteJavaS('com.example.MyTestTI', 'First', 'Second', 'Third');
```

### Expand

Expand développe le nom des variables TurboIntegrator encadrées par des signes % pour inclure leur valeur au cours de l'exécution. Cette fonction est fréquemment utilisée pour transmettre la valeur des variables TurboIntegrator à la fonction ODBCOutput.

Si le nom de la variable est une chaîne, la totalité de l'expression de variable doit être entre des guillemets anglais. Par exemple, "%V1%".

Si une valeur numérique est utilisée pour la fonction Expand, une conversion type implicite est effectuée et la valeur numérique est convertie en chaîne.

Cette chaîne inclut au maximum dix caractères. Si le nombre converti comporte moins de dix caractères, des espaces sont placés au début de la chaîne. Seuls les trois premiers caractères décimaux sont convertis. Par exemple, la valeur numérique 0.123456789 est convertie en chaîne "0.123".

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
Expand(String);
```

Argument	Description
String	Une chaîne qui inclut les noms de variable TurboIntegrator entre des %.

## Exemple

```
ODBCOutPut( 'TransData', Expand( 'INSERER DANS LES VALEURS VENTES ( MOIS, PRODUIT, VENTES )  
( "%V0%", "%V1%",%V2% )' ) );
```

Cet exemple illustre l'utilisation de la fonction Expand avec la fonction ODBCOutput. L'exemple insère des enregistrements dans une base de données relationnelles intitulée Ventes et comprenant 3 colonnes : Month, Product, and Sales.

La fonction Expand convertit les variables V0, V1 et V2 à leurs valeurs réelles dans la vue. Si l'on suppose que la première valeur se trouvant dans la vue est 123 456 et est définie par les éléments Jan et Contrôle.

```
Expand( 'INSERER DANS LES VALEURS VENTES ( MOIS, PRODUIT, VENTES ) ( "%V0%", "%V1%",%V2% )' )
```

devient

```
INSERETINTOSALES( MOIS, PRODUIT, VENTES ) VALEURS ( Jan, Contrôle, 123 456 )'
```

lors de l'exécution.

## FileExists

FileExists détermine si un fichier spécifié existe. La fonction renvoie 1 si le fichier existe et 0 s'il n'existe pas.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Syntaxe

```
FileExists(File);
```

Argument	Description
File	Nom d'un fichier. Si aucun chemin complet n'est spécifié, TM1 recherche le fichier dans le répertoire de données du serveur.

### Exemple

```
FileExists('C:\tm1s7\pdata\model.dim');
```

Cet exemple détermine si le fichier model.dim existe.

## LogOutput

LogOutput consigne un message dans le fichier `tm1server.log` lorsqu'une erreur d'un niveau de gravité spécifié est détectée dans un processus TurboIntegrator.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

### Prérequis

Pour activer la consignation des messages de TurboIntegrator, vous devez ajouter le débogueur TM1. TILogOutput au fichier `tm1-log.properties` et affecter au débogueur le niveau de votre choix. Par exemple, l'ajout de `TM1.TILogOutput=DEBUG` au fichier `tm1-log.properties` active la consignation pour tous les niveaux de gravité. Pour plus de détails sur le fichier `tm1-log.properties`, reportez-vous à la rubrique "Configuration et activation de la journalisation des messages de serveur" dans *TM1 Operations*.

## Syntaxe

```
LogOutput('SeverityLevel', 'MessageString');
```

Argument	Description
SeverityLevel	Niveau de gravité qui déclenche la consignation dans le fichier tm1server.log. Les valeurs valides pour cet argument sont les suivants : <ul style="list-style-type: none"><li>• 'DEBUG'</li><li>• 'INFO'</li><li>• 'WARN'</li><li>• 'ERROR'</li><li>• 'FATAL'</li></ul>
MessageString	Message à consigner dans le fichier tm1server.log. La chaîne du message peut être placée entre apostrophes ou correspondre à une autre fonction TurboIntegrator qui renvoie une chaîne.

## Exemples

```
LogOutput('WARN', 'TI process encountered a warning condition');
```

```
LogOutput('ERROR', TM1User() );
```

## TM1User

TM1User renvoie une chaîne indiquant le client TM1 en cours. Lorsqu'elle survient dans un processus exécuté directement par l'utilisateur, elle renvoie le nom de client TM1 de l'utilisateur. Lorsqu'elle survient dans une tâche exécutée directement par l'utilisateur, elle renvoie également le nom de client TM1 de l'utilisateur.

Lorsqu'elle est exécutée depuis une tâche planifiée, elle renvoie un nom sous la forme *R\*<nom de la tâche>* (par exemple, *R\*UpdateRegionDimension*).

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
TM1User()
```

## WildcardFileSearch

WildcardFileSearch permet d'utiliser des caractères génériques pour rechercher des fichiers dans un répertoire spécifié.

Les résultats de la fonction WildCardFileSearch varient selon le système d'exploitation utilisé. Les fichiers d'un répertoire Windows sont triés par ordre alphabétique, alors que les fichiers d'un répertoire UNIX sont triés dans un ordre aléatoire. Etant donné que l'ordre de tri varie d'un système d'exploitation à l'autre, la même fonction WildCardFileSearch exécutée sur des répertoires identiques, une sous Windows et une sous UNIX, produit des résultats différents.

Cette fonction est valide uniquement dans les processus TM1 TurboIntegrator.

## Syntaxe

```
WildcardFileSearch( Pathname, PriorFilename);
```



Argument	Description
Pathname	<p>Nom de chemin des fichiers à rechercher. Ce chemin doit se terminer par un nom de fichier, qui peut contenir une séquence de caractères génériques avec * et/ou ?.</p> <p>Le caractère générique ? correspond à n'importe quel caractère unique.</p> <p>Le caractère générique * correspond à aucun, un ou plusieurs caractères.</p>
PriorFilename	<p>Nom d'un fichier existant dans le répertoire spécifié. Ce nom de fichier ne doit pas contenir de caractères génériques. La recherche avec caractères génériques spécifiée par l'argument Pathname commence <i>APRES</i> ce fichier.</p> <p>Si vous passez une chaîne vide comme argument PriorFilename, la fonction WildcardFileSearch renvoie le premier fichier correspondant à la séquence avec caractères génériques spécifiée par l'argument Pathname.</p>

### Exemple

L'exemple suivant illustre l'utilisation de la fonction WildcardFileSearch afin de déterminer le premier fichier journal du serveur généré en 2004 :

```
file = WildcardFileSearch( 'C:\Program Files\Cognos\TM1\Custom\
TM1Data\SData\tm1s2004*.log', ' ');
```

Cet exemple renvoie le premier fichier correspondant à la séquence avec caractère générique 'tm1s2004\*.log' du répertoire C:\Program Files\Cognos\TM1\Custom\TM1Data\SData\.

Comme les fichiers journaux de serveur sont nommés et enregistrés avec des horodatages séquentiels et que le deuxième paramètre de WildcardFileSearch est vide, la fonction renvoie le premier fichier journal de serveur commençant par les caractères 'tm1s2004'. Ce fichier est le premier fichier journal de serveur généré en 2004.

L'exemple suivant illustre l'utilisation de la fonction WildcardFileSearch pour renvoyer le premier fichier journal du serveur généré après le fichier tm1s20040211153827.log :

```
file = WildcardFileSearch( 'C:\Program Files\Cognos\TM1\Custom\
TM1Data\SData\tm1s*.log', 'tm1s20040211153827.log
');
```

Cet exemple lance la recherche dans le répertoire C:\Program Files\Cognos\TM1\Custom\TM1Data\SData\ immédiatement après le fichier tm1s20040211153827.log et renvoie le premier fichier ultérieur correspondant à la séquence avec caractère générique 'tm1s\*.log'.

1s20040220175522.log est le premier fichier après tm1s20040211153827.log et correspondant à la séquence de caractères génériques. Par conséquent, l'exemple renvoie tm1s20040220175522.log.



# Chapitre 6. Variables TurboIntegrator

Les variables IBM Planning Analytics sont répertoriées ici par catégories.

## Variables locales TurboIntegrator

Quand vous exécutez un processus TurboIntegrator, un ensemble de variables locales implicites est généré. Les variables locales n'existent que dans le contexte du processus où elles sont utilisées et ne sont pas disponibles hors du processus. Les variables locales sont détruites à la sortie d'un processus. Ces variables, dont la liste suit, peuvent être remplacées pour manipuler un processus.

### DatasourceASCIIIDecimalSeparator

Cette variable locale TurboIntegrator définit le séparateur décimal à utiliser pour toute conversion de chaîne en nombre ou de nombre en chaîne. Si vous définissez cette variable, vous devez aussi définir la variable DatasourceASCIIIThousandSeparator.

Le caractère spécifié doit être un caractère imprimable ASCII standard, avec une valeur décimale comprise entre 33 et 127 inclus.

#### Syntaxe

```
DatasourceASCIIIDecimalSeparator=' Char' ;
```

ou

```
DatasourceASCIIIDecimalSeparator=Char(xx) ;
```

Argument	Description
Char	Caractère ASCII à utiliser comme séparateur. Le caractère peut être spécifié en tant que caractère entre apostrophes ou en tant que code décimal de caractère ASCII sans guillemets.

L'un ou l'autre des exemples ci-après définit la virgule (,) comme séparateur.

```
DatasourceASCIIIDecimalSeparator=', ' ;
```

```
DatasourceASCIIIDecimalSeparator=Char(44) ;
```

### DatasourceASCIIIDelimiter

Cette variable locale TurboIntegrator définit le caractère ASCII à utiliser comme délimiteur de zones quand DatasourceType a la valeur 'CHARACTERDELIMITED'.

Le caractère spécifié doit être un caractère imprimable ASCII standard, avec une valeur décimale comprise entre 33 et 127 inclus.

#### Syntaxe

```
DatasourceASCIIIDelimiter='Char' ;
```

ou

```
DatasourceASCIIDelimiter=Char(xx);
```

Argument	Description
Char	Caractère ASCII à utiliser comme délimiteur. Le caractère peut être spécifié en tant que caractère entre apostrophes ou en tant que code décimal de caractère ASCII sans guillemets.

L'un ou l'autre des exemples ci-après définit le tiret (-) comme délimiteur de zone.

```
DatasourceASCIIDelimiter='- ';
```

```
DatasourceASCIIDelimiter=Char(45);
```

### DatasourceASCIIHeaderRecords

Cette variable TurboIntegrator locale indique le nombre d'enregistrements à ignorer avant de traiter la source de données.

#### Syntaxe

```
DatasourceASCIIHeaderRecords=N;
```

Argument	Description
N	Nombre d'enregistrements à ignorer avant de traiter la source de données.

### DatasourceASCIIQuoteCharacter

Cette variable locale TurboIntegrator définit les caractères ASCII utilisés pour délimiter les zones du fichier source quand la valeur de DatasourceType est 'CHARACTERDELIMITED'.

Le caractère spécifié doit être un caractère imprimable ASCII standard, avec une valeur décimale comprise entre 33 et 127 inclus.

#### Syntaxe

```
DatasourceASCIIQuoteCharacter='Char';
```

ou

```
DatasourceASCIIQuoteCharacter=Char(xx);
```

Argument	Description
Char	Caractère ASCII délimitant les zones de la source de données. Le caractère peut être spécifié en tant que caractère entre apostrophes ou en tant que code décimal de caractère ASCII sans guillemets.

L'un ou l'autre des exemples ci-après définit l'astérisque (\*) comme délimiteur de zone.

```
DatasourceASCIIQuoteCharacter='*';
```

```
DatasourceASCIIQuoteCharacter=Char(42);
```

## DatasourceASCIIThousandsSeparator

Cette variable locale TurboIntegrator définit le séparateur de milliers à utiliser pour toute conversion de chaîne en nombre ou de nombre en chaîne.

Si vous définissez cette variable, vous devez aussi définir la variable DatasourceASCIIDecimalSeparator.

Le caractère spécifié doit être un caractère imprimable ASCII standard, avec une valeur décimale comprise entre 33 et 127 inclus.

### Syntaxe

```
DatasourceASCIIThousandsSeparator='Char';
```

ou

```
DatasourceASCIIThousandsSeparator=Char(xx);
```

Argument	Description
Char	Caractère ASCII à utiliser comme séparateur. Le caractère peut être spécifié en tant que caractère entre apostrophes ou en tant que code décimal de caractère ASCII sans guillemets.

L'un ou l'autre des exemples ci-après définit le point (.) comme séparateur de milliers.

```
DatasourceASCIIThousandsSeparator='.';
```

```
DatasourceASCIIThousandsSeparator=Char(46);
```

## DatasourceCubeview

Cette variable locale TurboIntegrator définit la vue à traiter si DatasourceType a la valeur 'VIEW'.

### Syntaxe

```
DatasourceCubeview='ViewName';
```

Argument	Description
ViewName	Nom de la vue à traiter. Il doit s'agir d'une vue de cube existante spécifiée par la variable DataSourceNameForServer.

## DatasourceDimensionSubset

Cette variable locale TurboIntegrator définit le sous-ensemble à traiter si DatasourceType a la valeur 'SUBSET'.

`DatasourceNameForServer=Nom dimension` est aussi nécessaire avec DATASOURCEDIMENSIONSUBSET pour que TM1 puisse identifier l'emplacement du sous-ensemble.

## Syntaxe

```
DatasourceDimensionSubset='SubsetName';
```

Argument	Description
SubsetName	Nom du sous-ensemble à traiter.

## DatasourceNameForServer

Cette variable locale TurboIntegrator définit le nom de la source de données (fichier .cma, nom de cube, source ODBC) utilisée par le serveur lors de l'exécution du processus.

### Syntaxe

```
DatasourceNameForServer='Name';
```

Argument	Description
Name	Pour une source de données .cma, le chemin d'accès complet du fichier .cma.  Pour les cubes, le nom du cube précédé de la chaîne 'local:'.  Pour une source ODBC, le nom de la source.

## DatasourceNameForClient

Cette variable locale TurboIntegrator définit le nom de la source de données (fichier .cma, nom de cube, source ODBC) utilisée par le client lors de la création ou de l'édition du processus.

### Syntaxe

```
DatasourceNameForClient='Name';
```

Argument	Description
Name	Pour une source de données .cma, le chemin d'accès complet du fichier .cma.  Pour les cubes, le nom du cube précédé de la chaîne 'local:'.  Pour une source ODBC, le nom de la source.

## DataSourceODBCCatalog

Cette variable locale TurboIntegrator définit le nom de la collection de base de données qui contient les cubes, les dimensions ou les autres objets auxquels vous souhaitez vous connecter. Dans Microsoft Analysis Services, il s'agit du nom de la base de données.

### Syntaxe

```
DataSourceODBCCatalog='Catalog';
```

Argument	Description
Catalog	Le nom de la collection de base de données à laquelle vous souhaitez vous connecter.

### DataSourceODBOConnectionString

Cette variable locale TurboIntegrator définit tout paramètre de connexion additionnel qui peut être requis pour la connexion au serveur OLAP.

#### Syntaxe

```
DataSourceODBOConnectionString='String';
```

Argument	Description
String	La valeur utilisée pour définir tout paramètre de connexion additionnel.  Affectez ces paramètres à cette variable en les séparant par des points-virgules.

### DataSourceODBOCubeName

Cette variable locale TurboIntegrator définit le nom du cube se trouvant sur le serveur OLAP que vous souhaitez utiliser comme source de données.

#### Syntaxe

```
DataSourceODBOCubeName='Name';
```

Argument	Description
Name	Le nom du cube à utiliser.

### DataSourceODBOHierarchyName

Cette variable locale TurboIntegrator définit le nom de la hiérarchie pour la dimension spécifique que vous utilisez comme source de données. Vous utilisez cette variable pour les autres produits OLAP, tels que SAP BW, où une hiérarchie est un objet distinct.

Cette variable n'est pas utilisée avec des sources de données TM1.

#### Syntaxe

```
DataSourceODBOHierarchyName='Name';
```

Argument	Description
Name	Nom de la hiérarchie pour une dimension spécifique.

## DataSourceODBOLocation

Cette variable locale TurboIntegrator définit le nom de l'emplacement (du système) sur lequel le serveur OLAP tourne.

TM1 utilise cette variable, mais les autres serveurs OLAP ne le font pas. Pour TM1, il s'agit de l'emplacement où l'hôte admin est exécuté.

### Syntaxe

```
DataSourceODBOLocation='Location';
```

Argument	Description
Location	Le nom de l'emplacement (du système) pour le serveur OLAP.

## DataSourceODBOProvider

Cette variable locale TurboIntegrator définit le nom du fournisseur ODBO que vous souhaitez utiliser comme source de données. Il s'agit du nom complet affecté par le fournisseur ODBO pour identifier son serveur de bases de données multidimensionnelles.

Vous devez utiliser le nom d'un fournisseur ODBO installé sur votre serveur.

### Syntaxe

```
DataSourceODBOProvider='Provider';
```

Argument	Description
Provider	Le nom du fournisseur ODBO à utiliser comme source de données.  Des noms de fournisseurs usuels sont :  TM1 OLE DB MD Provider  Microsoft OLE DB Provider for OLAP Services 8.0  SAP BW OLE DB Provider

## DataSourceODBOSAPClientID

Cette variable locale TurboIntegrator définit le numéro de client correspondant à la version de l'interface utilisateur qui se trouve sur le serveur SAP auquel vous souhaitez vous connecter.

### Syntaxe

```
DataSourceODBOSAPClientID='ID';
```

Argument	Description
ID	Un nombre correspondant à la version de l'interface utilisateur sur le serveur SAP.  Par exemple, 498.



## DataSourceODBSAPClientLanguage

Cette variable locale TurboIntegrator définit la spécification de langue du système SAP auquel vous souhaitez vous connecter.

### Syntaxe

```
DataSourceODBSAPClientLanguage='Language';
```

Argument	Description
Language	La spécification de langue du système SAP. Pour l'anglais des Etats-Unis, utilisez EN. Pour le français, utilisez FR. Pour les autres langues, consultez la documentation SAP.

## DatasourcePassword

Cette variable locale TurboIntegrator définit le mot de passe utilisé pour la connexion à la source de données.

### Syntaxe

```
DatasourcePassword='Password';
```

Argument	Description
Password	Le mot de passe utilisé pour la connexion à la source de données défini avec DatasourceNameForServer.

## DatasourceQuery

Cette variable locale TurboIntegrator définit la chaîne de requête à utiliser avec la source de données.

### Syntaxe

```
DatasourceQuery='Query';
```

Argument	Description
Query	Chaîne de requête à utiliser avec la source de données qui a été définie avec DatasourceNameForServer.

## DataSourceSAPUsingRoleAuths

La variable locale DataSourceSAPUsingRoleAuths demande au processus TurboIntegrator d'ignorer les informations de sécurité lors du traitement d'une source de données SAP. Cette variable doit être placée dans le prologue.

### Syntaxe

```
DataSourceSAPUsingRoleAuths='0'
```

Argument	Description
0	Les informations de sécurité sont <i>ignorées</i> lors du traitement d'une source de données SAP.
1	Les informations de sécurité sont <i>lues</i> lors du traitement d'une source de données SAP.

## DataSourceSAPUsingTexts

La variable locale DataSourceSAPUsingTexts demande au processus TurboIntegrator d'ignorer les descriptions de caractéristiques lors du traitement d'une source de données SAP afin de réduire la quantité de mémoire utilisée et d'augmenter les performances. Cette variable doit être placée dans le prologue.

### Syntaxe

```
DataSourceSAPUsingTexts='0'
```

Argument	Description
0	Les descriptions de caractéristiques sont <i>ignorées</i> lors du traitement d'une source de données SAP. Le nom technique de la caractéristique est importé dans TM1 à la fois en tant qu'alias et que nom d'élément.
1	Les descriptions de caractéristiques sont <i>lues</i> lors du traitement d'une source de données SAP.

## DatasourceType

Cette variable locale TurboIntegrator définit le type de la source de données.

### Syntaxe

```
DatasourceType='Type' ;
```

Argument	Description
Type	Les types valides sont les suivants : 'CHARACTERDELIMITED', 'POSITIONDELIMITED', 'VIEW', 'SUBSET', ODBC' et 'OLEDBOLAP', 'SAPCHARACTERISTICTEXTS'

## DatasourceUsername

Cette variable locale TurboIntegrator définit le nom utilisé pour la connexion à la source de données.

### Syntaxe

```
DatasourceUsername='Name' ;
```

Argument	Description
Name	Le nom utilisé pour la connexion à la source de données défini avec DataSourceNameForServer.

## MinorErrorLogMax

Cette variable locale TurboIntegrator définit le nombre d'erreurs mineures qui seront écrites dans le fichier TM1ProcessError.log pendant l'exécution du processus. Si cette variable n'est pas définie dans le processus, le nombre par défaut d'erreurs mineures écrites dans le fichier journal est 1000.

### Syntaxe

```
MinorErrorLogMax=N;
```

Argument	Description
N	<p>Valeur indiquant le nombre d'erreurs devant être écrites dans le fichier journal.</p> <p>Spécifiez un entier supérieur à zéro pour définir le nombre maximum d'erreurs écrites dans le fichier journal.</p> <p>Spécifiez la valeur 0 pour ne consigner aucune erreur lors de l'exécution du processus.</p> <p>Spécifiez la valeur -1 pour permettre la journalisation d'un nombre illimité d'erreurs mineures dans le fichier journal.</p>

Le tableau suivant fournit un exemple de message de journal des erreurs et du résultat correspondant.

Exemple	Résultat
MinorErrorLogMax=750;	Le fichier journal accepte jusqu'à 750 erreurs.
MinorErrorLogMax=0;	Aucune erreur n'est écrite dans le fichier journal.
MinorErrorLogMax=-1;	Un nombre illimité d'erreurs peuvent être écrites dans le fichier journal.

## NValue

Quand la valeur de DataSourceType est 'VIEW', cette variable locale TurboIntegrator détermine la valeur de la cellule quand Value\_Is\_String est 0 (c'est-à-dire quand la cellule en cours est numérique).

### Syntaxe

```
Nvalue=N;
```

Argument	Description
N	Valeur de la cellule en cours.

## OnMinorErrorDoItemSkip

Cette variable locale TurboIntegrator donne à TurboIntegrator l'instruction d'ignorer l'enregistrement suivant quand une erreur mineure est détectée lors du traitement d'un enregistrement.

Cette variable est utile lorsqu'une zone ou une valeur incorrecte d'un enregistrement produit plusieurs erreurs mineures.

Par exemple, si vous avez 100 fonctions CELLPUTN dans un processus et qu'une des zones d'un enregistrement donné est 'erronée' ou invalide, le nombre d'erreurs mineures est incrémenté de 100, (1 pour chaque fonction CELLPUTN qui rencontre l'erreur.) Ces 100 erreurs mineures sont comptées dans la limite d'erreurs mineures définie par MinorErrorLogMax. Un processus TurboIntegrator échoue s'il dépasse le nombre d'erreurs mineures défini par MinorErrorLogMax.

Si OnMinorErrorDoItemSkip=1; est inclus dans l'onglet Prologue du processus, celui-ci ignore immédiatement l'enregistrement suivant lorsqu'une zone 'erronée' ou invalide est détectée dans un enregistrement source. Dans l'exemple ci-dessus, il en résulte que le nombre d'erreurs mineures est incrémenté d'une unité seulement, au lieu de 100.

### Syntaxe

```
OnMinorErrorDoItemSkip=N;
```

Argument	Description
N	Valeur indiquant si l'enregistrement doit être ignoré lorsqu'une erreur mineure est détectée.  1 (ou toute autre valeur non nulle) donne l'instruction d'ignorer l'enregistrement quand une erreur mineure est détectée.  0 indique que TurboIntegrator doit continuer de traiter l'enregistrement en cours quand une erreur mineure se produit.

## SValue

Quand la valeur de DatasourceType est 'VIEW', cette variable locale TurboIntegrator détermine la valeur de la cellule quand Value\_Is\_String n'est pas 0, (quand la cellule en cours contient une chaîne).

### Syntaxe

```
Svalue='String';
```

Argument	Description
String	Valeur de la cellule en cours.

## Fichier TM1ProcessError.log

Quand un processus TurboIntegrator rencontre une erreur, il génère un fichier TM1ProcessError.log. Ce fichier journal est enregistré dans le répertoire de données du serveur où réside le processus.

Un fichier TM1ProcessError.log contient une liste des erreurs rencontrées par le processus. Pour chaque erreur rencontrée, le fichier journal enregistre l'onglet et la ligne à l'origine de l'erreur, avec une brève description de celle-ci.

Lorsqu'un fichier journal des erreurs de processus est généré, TM1 affecte un nom unique qui permet d'identifier facilement le processus TurboIntegrator à l'origine du fichier d'erreurs, ainsi que l'heure de création du fichier. Les noms de fichiers sont affectés à l'aide de la convention suivante :

TM1ProcessError\_<horodatage>\_<nom du processus>.log.

Dans cette convention, <horodatage> est l'heure (exprimée en format aaaammjjhhmmss GMT) à laquelle le fichier a été généré et <nom processus> est le nom du processus TurboIntegrator à l'origine des erreurs.

Par exemple, un fichier d'erreurs appelé TM1ProcessError\_20040224203148\_CreateSalesCube.log indique que le fichier d'erreurs a été créé à 20:31:48 GMT le 24.02.04 et qu'il contient des erreurs créées par le processus CreateSalesCube.

Une intervention manuelle est nécessaire pour supprimer ou archiver ces fichiers journaux. Un nouveau fichier journal est généré chaque fois qu'un processus TurboIntegrator comporte une erreur (1 fichier journal par exécution TurboIntegrator).

De nombreux journaux d'erreurs de processus TurboIntegrator peuvent être générés pour les processus TurboIntegrator qui s'exécutent fréquemment et qui génèrent une erreur à chaque exécution. De nombreux fichiers journaux générés dans le répertoire des journaux de la base de données TM1 peuvent avoir une incidence sur les performances de la base de données TM1 lorsqu'elle crée ou met à jour des fichiers journaux.

Un trop grand nombre de fichiers dans le répertoire des journaux de la base de données TM1 peut affecter le téléchargement des fichiers journaux du composant Administration Planning Analytics. Il est recommandé d'avoir moins de 2000 fichiers dans le répertoire de journalisation de la base de données TM1 .

## Value\_Is\_String

Quand le type de source de données est 'VIEW', cette variable locale TurboIntegrator détermine si la cellule en cours doit être traitée comme chaîne ou valeur numérique.

### Syntaxe

```
Value_Is_String=N;
```

Argument	Description
N	Valeur indiquant si la cellule en cours est une chaîne ou une valeur numérique.  0 signifie que la cellule est un nombre, toute autre valeur signifie que la cellule est traitée comme une chaîne.

## Variables globales TurboIntegrator

Ce type de variable TurboIntegrator est associé à une tâche ou un processus TM1 individuel et tout sous-processus attendant. Il existe deux types de variables globales : implicites et définies par l'utilisateur. Les variables globales implicites sont décrites ici. Les variables globales définies par l'utilisateur sont décrites plus bas.

Les variables globales peuvent être utilisées de deux façons :

- Les variables globales peuvent être déclarées dans un processus faisant partie d'une tâche donnée. Une fois déclarées, les variables globales sont disponibles pour tous les autres processus qui font partie de la tâche. Ces variables persistent pendant l'exécution de la tâche et la durée de la session de serveur en cours. Les variables globales sont détruites lors de l'arrêt du serveur.
- Les variables globales peuvent être déclarées dans un processus et mises à la disposition de tout processus ultérieur appelé par la fonction ExecuteProcess(). Ces sous-processus doivent utiliser les mêmes instructions de déclaration de variable globale (décrites ci-dessous) pour accéder aux variables globales.

Si un nom de variable globale est identique à un nom de variable locale, la définition de variable locale a priorité et remplace celle de la variable globale.

Les variables globales sont déclarées dans un processus TurboIntegrator à l'aide d'une des deux fonctions suivantes :

### **NumericGlobalVariable('NomVariable');**

Permet de déclarer une variable globale numérique.

### **StringGlobalVariable('NomVariable');**

Permet de définir une variable globale chaîne.

## **Variables globales implicites**

---

Quand vous exécutez un processus TurboIntegrator, un ensemble de variables globales implicites est généré. Si le processus générant les variables fait partie d'une tâche, ces variables globales sont disponibles et peuvent être partagées par tous les autres processus qui font partie de la tâche.

De plus, toutes les variables globales implicites d'un processus sont disponibles et partagées par les processus ultérieurs appelés par la fonction `ExecuteProcess()`.

Bien que les variables implicites soient générées par le processus TurboIntegrator, vous devez déclarer une variable pour pouvoir l'utiliser dans un processus.

Les variables globales implicites sont déclarées dans un processus TurboIntegrator à l'aide de la fonction `NumericGlobalVariable('VariableName');` :

Cliquez sur un lien ci-dessous pour des détails sur une variable globale implicite.

- [DataMinorErrorCount](#).
- [MetadataMinorErrorCount](#).
- [ProcessReturnCode](#).
- [PrologMinorErrorCount](#).

Par exemple, pour utiliser la variable globale implicite `PrologMinorErrorCount` dans un processus, vous devez d'abord la déclarer comme suit :

```
NumericGlobalVariable('PrologMinorErrorCount');
```

### **DataMinorErrorCount**

Cette variable globale TurboIntegrator compte les erreurs mineures qui se produisent dans la portion Données d'un processus TurboIntegrator. Pour chaque erreur mineure rencontrée, la valeur de la variable est incrémentée de 1.

#### **Syntaxe**

```
DataMinorErrorCount=N;
```

<b>Argument</b>	<b>Description</b>
N	Nombre d'erreurs mineures rencontrées dans la partie Données du processus.

## MetadataMinorErrorCount

Cette variable globale TurboIntegrator compte les erreurs mineures qui se produisent dans la portion Métadonnées d'un processus TurboIntegrator. Pour chaque erreur mineure rencontrée, la valeur de la variable est incrémentée de 1.

### Syntaxe

```
MetadataMinorErrorCount=N;
```

Argument	Description
N	Nombre d'erreurs mineures rencontrées dans la partie Métadonnées du processus.

## ProcessReturnCode

Cette variable globale TurboIntegrator stocke le statut de sortie du dernier processus TurboIntegrator exécuté.

### Syntaxe

```
ProcessReturnCode=StatusCode;
```

Code de statut	Description
ProcessExitByBreak()	Indique la sortie du processus sur détection de fonction ProcessBreak.
ProcessExitByChoreQuit()	Indique que le processus s'est fermé en raison de l'exécution de la fonction ChoreQuit.
ProcessExitByChoreRollback()	Indique la sortie du processus sur détection de fonction ChoreRollback.
ProcessExitByProcessRollback()	Indique la sortie du processus sur détection de fonction ProcessRollback.
ProcessExitByQuit()	Indique la sortie du processus en raison d'une commande quit explicite.
ProcessExitMinorError()	Indique que l'exécution du processus a abouti mais que des erreurs mineures sont survenues.
ProcessExitNormal()	Indique que le processus s'est exécuté normalement.
ProcessExitOnInit()	Indique que le processus a été abandonné au cours de l'initialisation.
ProcessExitSeriousError()	Indique la sortie du processus en raison d'une erreur grave.
ProcessExitWithMessage()	Indique la sortie normale du processus, avec un message écrit dans tm1server.log.

## PrologMinorErrorCount

Cette variable globale TurboIntegrator compte les erreurs mineures qui se produisent dans la portion Prologue d'un processus TurboIntegrator. Pour chaque erreur mineure rencontrée, la valeur de la variable est incrémentée de 1.

### Syntaxe

```
PrologMinorErrorCount=N;
```

Argument	Description
N	Nombre d'erreurs mineures rencontrées dans le Prologue.

## Variables utilisateur TurboIntegrator

Ce type de variable est associé à un utilisateur TM1 individuel, et non à un processus ou une tâche particulière. Les variables utilisateur peuvent être manipulées depuis un processus ou une tâche TurboIntegrator alors que l'utilisateur auquel la variable est associée est connecté au serveur.

Les variables utilisateur doivent être déclarées explicitement. Une fois déclarées, elles persistent pendant toute la durée de la session TM1 de l'utilisateur (jusqu'à ce que celui-ci se déconnecte ou soit déconnecté du serveur).

Les variables utilisateur sont déclarées dans un processus TurboIntegrator à l'aide d'une des deux fonctions suivantes :

- `NumericGlobalVariable('VariableName');` Permet de déclarer une variable utilisateur numérique.
- `StringGlobalVariable('VariableName');` Permet de définir une variable utilisateur chaîne.

Les variables utilisateur sont créées la première fois qu'une telle déclaration est détectée dans un processus TurboIntegrator actif.

Une fois celle-ci créée, le nom de la variable peut être référencé et utilisé comme toute autre variable locale ou globale, mais sa valeur ne persiste dans les processus et les tâches que pendant que le créateur de la variable reste connecté au serveur.



## Mentions légales

---

Le présent document concerne des produits et des services disponibles dans différents pays.

Il peut être mis à disposition par IBM dans d'autres langues. Toutefois, il peut être nécessaire de posséder une copie du produit ou de la version du produit dans cette langue pour pouvoir y accéder.

Il peut contenir des informations ou des références concernant certains produits, logiciels ou service IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM. Le présent document peut décrire des produits, des services ou des fonctions qui ne sont pas inclus dans le Logiciel ni dans l'autorisation d'utilisation que vous avez acquise.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7 Canada

Les informations sur les licences concernant les produits IBM utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font

pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Software Group  
Attention: Licensing  
3755 Riverside Dr.  
Ottawa, ON  
K1V 1B7  
Canada

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret contractuel IBM, des Conditions internationales d'utilisation de logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

#### LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Ces exemples de programmes sont fournis "en l'état", sans garantie d'aucune sorte. IBM n'est en aucun cas responsable des dommages liés à l'utilisation de ces exemples de programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. \_entrer l'année ou les années\_.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Cette Offre Logiciels n'utilise pas de cookies ou d'autres techniques pour collecter des informations personnelles identifiables.



---

## Informations sur le produit

Ce document s'applique à IBM Planning Analytics version 2.0.0 et peut aussi s'appliquer aux éditions ultérieures.

## Copyright

Licensed Materials - Property of IBM. Eléments sous licence - Propriété d'IBM

© Copyright IBM Corp. 2007, 2019.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques IBM est disponible sur la page Web "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" à [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Les noms suivants sont des marques d'autres sociétés :

- Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.
- Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans d'autres pays.
- La marque Linux est utilisée conformément à une sous-licence de Linux Foundation, le licencié exclusif de Linus Torvalds, propriétaire de la marque Linux dans le monde.
- UNIX est une marque de The Open Group aux Etats-Unis et/ou dans certains autres pays.
- Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.
- Red Hat, JBoss, OpenShift, Fedora, Hibernate, Ansible, CloudForms, RHCA, RHCE, RHCSA, Ceph et Gluster sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Les captures d'écran des produits Microsoft ont été utilisées avec l'autorisation de Microsoft.



# Index

## A

ABS [150](#)  
accès  
    privileges, affectations de sécurité [51](#)  
ACOS [150](#)  
AddClient [359](#)  
AddCubeDependency [282](#)  
AddGroup [359](#)  
AddInfoCubeRestriction [411](#)  
Admin  
    affectations de sécurité [54](#)  
    protocole TSL du serveur, options TM1 [74](#)  
affectation  
    affectations de sécurité, grille [51](#)  
aperçu de grille  
    Editeur TurboIntegrator [92](#)  
apparence du bouton d'action [4](#)  
application  
    Explorateur de serveur [59](#)  
ARRONDI [154](#)  
ASCII  
    Fonctions TurboIntegrator pour texte [217](#)  
ASCIIDelete [218](#)  
ASCIIOutput [218](#)  
ASIN [150](#)  
AssignClientPassword [360](#)  
AssignClientToGroup [359](#)  
AssociateCAMIDToGroup [360](#)  
ATAN [151](#)  
AttrDelete [228](#)  
attribut  
    Editeur [7](#)  
    Editeur TurboIntegrator [94](#)  
    fonctions TurboIntegrator de manipulation [226](#)  
AttrInsert [229](#)  
ATTRN [104](#)  
ATTRNL [226](#)  
AttrPutN [229](#)  
AttrPutS [230](#)  
ATTRS [105](#)  
AttrSL [227](#)  
aucun  
    affectations de sécurité [51](#)  
avancé  
    mappage de grille [5](#)  
    onglet de l'Editeur TurboIntegrator [98](#)  
    options [5](#)

## B

bac à sable, fonctions [348](#)  
barre d'outils [48](#)  
barre de statut [48](#)  
BatchUpdateFinish [291](#), [372](#)  
BatchUpdateFinishWait [373](#)  
BatchUpdateStart [374](#)

boîte de dialogue Alias TM1 [74](#)  
boîte de dialogue Choisir une vue (Navigateur de feuille de calcul) [36](#)  
boîte de dialogue Options TM1 [74](#)  
boîte de dialogue Supprimer les sous-ensembles nommés [24](#)  
boîte de dialogue Supprimer les vues nommées [25](#)  
boîtes de dialogue [1](#)  
bouton d'action  
    propriétés [1](#)  
boutons  
    Editeur TurboIntegrator [92](#)

## C

CAPIT [157](#)  
CellGetN [283](#)  
CellGetS [284](#)  
CellIncrementN [285](#)  
CellIsUpdateable [285](#)  
CellPutN [286](#)  
CellPutProportionalSpread [287](#)  
CellPutS [287](#)  
CellSecurityCubeCreate [361](#)  
CellSecurityCubeDestroy [361](#)  
CellValueN [115](#)  
CellValueS [116](#)  
CHAR [157](#)  
Chinois [47](#)  
ChoreAttrDelete [231](#)  
ChoreAttrInsert [231](#)  
ChoreAttrN [232](#)  
ChoreAttrNL [232](#)  
ChoreAttrPutN [233](#)  
ChoreAttrPutS [234](#)  
ChoreAttrS [235](#)  
ChoreAttrSL [235](#)  
ChoreError [281](#)  
ChoreRollback [281](#)  
Clients  
    boîte de dialogue Centre de messagerie [18](#)  
    fenêtre Groupes [15](#)  
    grille Groupes [15](#), [17](#)  
    menu Clients/Groupes [16](#)  
CODE [158](#)  
commentaires [47](#)  
comparaison [103](#)  
ConsolidatedAvg [109](#)  
ConsolidatedCount [112](#)  
ConsolidatedCountUnique [113](#)  
ConsolidatedMax [113](#)  
ConsolidatedMin [114](#)  
consolidation  
    Editeur TurboIntegrator [94](#)  
contention de verrouillage [376](#)  
CONTINUE [148](#)  
contrôle

contrôle (*suite*)  
   objets [48](#)  
   options [49](#)  
 Coréen [47](#)  
 COS [151](#)  
 CreateHierarchyByAttribute [236](#)  
 créer  
   cube, boîte de dialogue [19](#)  
   dimension, boîte de dialogue [18](#)  
   objet de répllication de serveur [19](#)  
 cube  
   boîte de dialogue Propriétés [21](#)  
   Editeur de sous-ensemble d'informations [44](#)  
   Editeur TurboIntegrator [94](#)  
   Explorateur de serveur [60](#)  
   fonctions de manipulation TurboIntegrator [282](#)  
   optimisation [20](#)  
   Viewer [21](#)  
 CubeAttrDelete [237](#)  
 CubeAttrInsert [237](#)  
 CubeATTRN [106](#)  
 CubeATTRNL [240](#)  
 CubeAttrPutN [238](#)  
 CubeAttrPutS [239](#)  
 CubeATTRS [106](#)  
 CubeATTRSL [241](#)  
 CubeClearData [288](#)  
 CubeCreate [289](#)  
 CubeDataReservationAcquire [294](#)  
 CubeDataReservationGet [296](#)  
 CubeDataReservationGetConflicts [298](#)  
 CubeDataReservationRelease [295](#)  
 CubeDataReservationReleaseAll [295](#)  
 CubeDestroy [289](#)  
 CubeDimensionCountGet [289](#)  
 CubeDRAcquire [299](#)  
 CubeDRGet [301](#)  
 CubeDRGetConflicts [303](#)  
 CubeDRRelease [300](#)  
 CubeDRReleaseAll [300](#)  
 CubeExists [290](#)  
 CubeGetLogChanges [290](#)  
 CubeProcessFeeders [345](#)  
 CubeRuleAppend [345](#)  
 CubeRuleDestroy [346](#)  
 CubeSetConnParams [292](#)  
 CubeSetLogChanges [292](#)  
 CubeTimeLastUpdated [293](#)  
 CubeUnload [293](#)

**D**

D\_FSAVE [168](#)  
 D\_PICK [167](#)  
 D\_SAVE [169](#)  
 DataMinorErrorCount [430](#)  
 DatasourceASCIIIDecimalSeparator [419](#)  
 DatasourceASCIIIDelimiter [419](#)  
 DatasourceASCIIHeaderRecords [420](#)  
 DatasourceASCIIQuoteCharacter [420](#)  
 DatasourceASCIIThousandSeparator [421](#)  
 DatasourceCubeview [421](#)  
 DatasourceDimensionSubset [421](#)  
 DatasourceNameForClient [422](#)  
 DatasourceNameForServer [422](#)  
 DataSourceODBCatalog [422](#)  
 DataSourceODBCConnectionString [423](#)  
 DataSourceODBCubeName [423](#)  
 DataSourceODBOHierarchyName [423](#)  
 DataSourceODBOLocation [424](#)  
 DataSourceODBOProvider [424](#)  
 DataSourceODBOSAPClientID [424](#)  
 DataSourceODBOSAPClientLanguage [425](#)  
 DatasourcePassword [425](#)  
 DatasourceQuery [425](#)  
 DataSourceSAPUsingRoleAuths [425](#)  
 DataSourceSAPUsingTexts [426](#)  
 DatasourceType [426](#)  
 DatasourceUsername [426](#)  
 DATE [121](#)  
 date et heure  
   fonctions TurboIntegrator [304](#)  
 DATES [121](#)  
 DAY [122](#)  
 DAYNO [122](#)  
 DBProportionalSpread [169](#)  
 DBR [189](#)  
 DBRA [190](#)  
 DBRW [191](#)  
 DBS [192](#)  
 DBSA [192](#)  
 DBSS [193](#)  
 DBSW [194](#)  
 décommenter [47](#)  
 DELET [158](#)  
 DeleteAllPersistentFeeders [347](#)  
 DeleteClient [362](#)  
 DeleteGroup [362](#)  
 Détails du journal d'audit, fenêtre [13](#)  
 DFRST [195](#)  
 dimension  
   boîte de dialogue Insertion d'élément [29](#)  
   boîte de dialogue Ordre des éléments [29](#)  
   Editeur de dimension, menu [25](#)  
   Editeur de sous-ensemble d'informations [45](#)  
   Editeur TurboIntegrator [94](#)  
   Explorateur de serveur [62](#)  
   fonctions de règles d'informations [128](#)  
   fonctions TurboIntegrator de manipulation [306](#)  
   Propriétés d'éléments, boîte de dialogue [30](#)  
 DimensionAttrDelete [242](#)  
 DimensionAttrInsert [242](#)  
 DimensionATTRN [106](#)  
 DimensionATTRNL [244](#)  
 DimensionAttrPutN [243](#)  
 DimensionAttrPutS [244](#)  
 DimensionATTRS [107](#)  
 DimensionATTRSL [245](#)  
 DimensionCreate [306](#)  
 DimensionDeleteAllElements [306](#)  
 DimensionDeleteElements [307](#)  
 DimensionDestroy [307](#)  
 DimensionElementComponentAdd [308](#)  
 DimensionElementComponentAddDirect [308](#)  
 DimensionElementComponentDelete [309](#)  
 DimensionElementComponentDeleteDirect [309](#)  
 DimensionElementDelete [310](#)  
 DimensionElementDeleteDirect [311](#)

[DimensionElementExists 311](#)  
[DimensionElementInsert 312](#)  
[DimensionElementInsertDirect 313, 318](#)  
[DimensionElementPrincipalName 314](#)  
[DimensionExists 314](#)  
[DimensionHierarchyCreate 315](#)  
dimensions en colonne  
    [Cube Viewer 21](#)  
dimensions en titre  
    [Cube Viewer 21](#)  
[DimensionSortOrder 315](#)  
[DimensionTimeLastUpdated 317](#)  
[DimensionTopElementInsert 317](#)  
[DimensionUpdateDirect 319](#)  
[DIMIX 129, 195](#)  
[DIMNM 129, 196](#)  
[DIMSIZ 129, 196](#)  
[DisableBulkLoadMode 374](#)  
[DisableMTQViewConstruct 396](#)  
divers  
    fonctions de règles [164](#)  
    fonctions TurboIntegrator [411](#)  
[DNEXT 130, 196](#)  
[DNLEV 130, 197](#)  
données  
    Editeur TurboIntegrator [94, 98](#)  
    source, onglet de l'Editeur TurboIntegrator [78](#)  
[DTYPE 131, 198](#)

## E

[E\\_PICK 170](#)  
écriture  
    affectations de sécurité [52](#)  
éditer  
    boîte de dialogue formule [31](#)  
    boîte de dialogue Référence au cube [32](#)  
Editeur [77](#)  
[ELCOMP 132, 198](#)  
[ELCOMPN 132, 199](#)  
élément  
    Editeur de dimension, sous-fenêtre [25](#)  
    éditeur de sous-ensemble, sous-fenêtre [68](#)  
    fonctions de règles d'informations [132](#)  
[ElementAttrDelete 251](#)  
[ElementAttrInsert 250](#)  
[ElementAttrN 107](#)  
[ElementATTRNL 246](#)  
[ElementAttrPutN 248](#)  
[ElementAttrPutS 249](#)  
[ElementAttrS 108](#)  
[ElementATTRSL 247](#)  
[ElementComponent 133](#)  
[ElementComponentCount 133](#)  
[ElementCount 134](#)  
[ElementFirst 134](#)  
[ElementIndex 135](#)  
[ElementIsAncestor 135](#)  
[ElementIsComponent 136](#)  
[ElementIsParent 136](#)  
[ElementLevel 137](#)  
[ElementName 138](#)  
[ElementNext 138](#)  
[ElementParent 139](#)

[ElementParentCount 139](#)  
[ElementSecurityGet 363](#)  
[ElementSecurityPut 363](#)  
[ElementType 140](#)  
[ElementWeight 140](#)  
élévation à une puissance [103](#)  
[ELISANC 141](#)  
[ELISCOMP 141, 199](#)  
[ELISPAR 142, 200](#)  
[ELLEV 142, 200](#)  
[ELPAR 143, 201](#)  
[ELPARN 143, 202](#)  
[ELSEN 202](#)  
[ELWEIGHT 144, 203](#)  
[EnableBulkLoadMode 375](#)  
[EnableMTQViewConstruct 397](#)  
enregistrer  
    boîte de dialogue Vue [50](#)  
    boîte de dialogue Vue du Navigateur de feuille de calcul [50](#)  
    sous-ensemble, boîte de dialogue [49](#)  
[ENT 152](#)  
épilogue  
    Editeur TurboIntegrator [98](#)  
Excel  
    fonctions macro [167](#)  
[ExecuteCommand 336](#)  
[ExecuteJavaN 412](#)  
[ExecuteJavaS 413](#)  
[ExecuteProcess 337, 429](#)  
[EXP 151](#)  
[Expand 414](#)

## F

[FEEDERS 164](#)  
[FEEDSTRINGS 164](#)  
fenêtre Journal d'audit [9](#)  
fenêtres et boîtes de dialogue [1](#)  
feuille de calcul  
    bouton d'action [3](#)  
[FileExists 415](#)  
[FILL 159](#)  
filtrer  
    les éléments par attribut, boîte de dialogue [32](#)  
    les éléments par niveau, boîte de dialogue [32](#)  
    sous-ensemble, boîte de dialogue [33](#)  
    vue, boîte de dialogue [34](#)  
fonctions  
    règles [103, 167](#)  
    TurboIntegrator [217](#)  
fonctions de calcul [109](#)  
fonctions de feuille de calcul TM1  
    [TM1ELLIST 205](#)  
    [TM1INFO 208](#)  
    [TM1PRIMARYDBNAME 209](#)  
Fonctions de feuille de calcul TM1  
    [TM1GLOBALSANDBOX 208](#)  
Fonctions de règles de hiérarchie [147](#)  
fonctions de règles de texte [157](#)  
fonctions de règles financières [145](#)  
fonctions de règles mathématiques [150](#)  
fonctions macro  
    accéder [167](#)

fonctions macro (*suite*)  
liste [167](#)  
fonctions TurboIntegrator de réservation de données  
CubeDataReservationAcquire [294](#)  
CubeDataReservationGet [296](#)  
CubeDataReservationGetConflicts [298](#)  
CubeDataReservationRelease [295](#)  
CubeDataReservationReleaseAll [295](#)  
CubeDRAcquire [299](#)  
CubeDRGet [301](#)  
CubeDRGetConflicts [303](#)  
CubeDRRelease [300](#)  
CubeDRReleaseAll [300](#)  
fonctions TurboIntegrator ODBC [333](#)  
ForceSkipCheck [347](#)  
FormatDate [304](#)  
Formule de variables de processus [94](#)

## G

GetProcessErrorFileDirectory [338](#)  
GetProcessErrorFilename [338](#)  
GetProcessName [339](#)  
GetUseActiveSandboxProperty [348](#)  
grille  
Editeur TurboIntegrator [92](#)  
groupe de serveurs  
Explorateur de serveur [56](#)

## H

hiérarchie  
fonctions de manipulation TurboIntegrator [319](#)  
HierarchyATTRN [253](#)  
HierarchyATTRNL [254](#)  
HierarchyAttrPutN [251](#)  
HierarchyAttrPutS [252](#)  
HierarchyATTRS [253](#)  
HierarchyATTRSL [255](#)  
HierarchyContainsAllLeaves [319](#)  
HierarchyCreate [320](#)  
HierarchyDeleteAllElements [320](#)  
HierarchyDeleteElements [321](#)  
HierarchyDestroy [321](#)  
HierarchyElementComponentAdd [321](#)  
HierarchyElementComponentAddDirect [322](#)  
HierarchyElementComponentDelete [323](#)  
HierarchyElementComponentDeleteDirect [323](#)  
HierarchyElementDelete [324](#)  
HierarchyElementDeleteDirect [325](#)  
HierarchyElementExists [326](#)  
HierarchyElementInsert [326](#)  
HierarchyElementInsertDirect [327](#)  
HierarchyElementPrincipalName [328](#)  
HierarchyElementSecurityGet [364](#)  
HierarchyElementSecurityPut [364](#)  
HierarchyExists [329](#)  
HierarchyHasOrphanedLeaves [329](#)  
HierarchySortOrder [330](#)  
HierarchySubsetAliasSet [378](#)  
HierarchySubsetAttrDelete [262](#)  
HierarchySubsetAttrInsert [261](#)  
HierarchySubsetATTRN [256](#)

HierarchySubsetATTRNL [258](#)  
HierarchySubsetAttrPutN [260](#)  
HierarchySubsetAttrPutS [259](#)  
HierarchySubsetATTRS [256](#)  
HierarchySubsetATTRSL [257](#)  
HierarchySubsetCreate [378](#)  
HierarchySubsetDeleteAllElements [379](#)  
HierarchySubsetDestroy [379](#)  
HierarchySubsetElementDelete [380](#)  
HierarchySubsetElementExists [380](#)  
HierarchySubsetElementGetIndex [381](#)  
HierarchySubsetElementInsert [382](#)  
HierarchySubsetExists [382](#)  
HierarchySubsetGetElementName [383](#)  
HierarchySubsetGetSize [383](#)  
HierarchySubsetIsAllSet [384](#)  
HierarchySubsetMDXGet [384](#)  
HierarchySubsetMDXSet [384](#)  
HierarchyTimeLastUpdated [331](#)  
HierarchyTopElementInsert [331](#)  
HierarchyTopElementInsertDirect [332](#)  
HierarchyUpdateDirect [333](#)

## I

I\_EXPORT [171](#)  
I\_NAMES [172](#)  
I\_PROCESS [172](#)  
If [339](#)  
IF [149](#)  
ignorer les paramètres  
extraction de vue [100](#)  
importer [46](#)  
Impression de rapports, assistant [39](#)  
Imprimer [46](#)  
infobulles [48](#)  
insérer une référence de cube [49](#)  
INSRT [159](#)  
ISUND [152](#)  
ISUNDEFINEDCELLVALUE [118](#)  
ItemReject [340](#)  
ItemSkip [340](#)

## J

Japonais [47](#)  
jeu de caractères [220](#)  
jeu de caractères de grande taille [47](#)  
journal d'audit  
fenêtre [9](#)  
fenêtre de détails [13](#)  
journal de messages  
fenêtre [37](#)  
journal des messages, fenêtre [37](#)  
Journal des messages, fenêtre [37](#)

## K

KEY\_ER [189](#)

## L

lecture



lecture (*suite*)  
affectations de sécurité [52](#)  
LevelCount [144](#)  
ligne  
Cube Viewer [21](#)  
LN [152](#)  
LOG [153](#)  
logique  
fonctions de règles [148](#)  
opérateurs [103](#)  
LONG [159](#)  
LOWER [160](#)

## M

M\_CLEAR [173](#)  
mappage automatique 5  
mappages, onglet de l'Éditeur TurboIntegrator [94](#)  
MAX [153](#)  
menu aide  
Fenêtre Journal des messages [38](#)  
menu dynamique  
Explorateur de serveur [56](#)  
menu Edition  
Attributs [7](#)  
Cube Viewer [22](#)  
Editeur de dimension [25](#)  
Editeur de sous-ensemble [69](#)  
Editeur TurboIntegrator [78](#)  
Explorateur de serveur [67](#)  
Fenêtre Journal des messages [38](#)  
Résultats de la requête du journal des transactions [76](#)  
menu fichier  
Attributs [7](#)  
Cube Viewer [22](#)  
Editeur TurboIntegrator [77](#)  
Explorateur de serveur [55](#)  
Fenêtre Journal des messages [37](#)  
menu groupes  
Clients/Groupes [16](#)  
menu Navigateur de feuille de calcul [36](#)  
menu Outils  
Editeur de sous-ensemble [73](#)  
menu vue  
Cube Viewer [23](#)  
Editeur de dimension [28](#)  
Editeur de sous-ensemble [72](#)  
Explorateur de serveur [68](#)  
MetadataMinorErrorCount [431](#)  
métadonnées  
Editeur TurboIntegrator [98](#)  
méthode Exécuter [167](#)  
mettre en retrait [47](#)  
MIN [153](#)  
MinorErrorLogMax [427](#)  
MOD [154](#)  
modules VBA  
fonctions macro [167](#)  
MONTH [123](#)

## N

N\_CONNECT [173](#)

N\_DISCONNECT [173](#)  
NewDateFormatter [304](#)  
nouvel attribut, boîte de dialogue [38](#)  
NOW [123](#)  
NumberToString [219](#)  
NumberToStringEx [220](#)  
NUMBR [160](#)  
NumericGlobalVariable(NomVariable) [430](#)  
NumericSessionVariable(NomVariable) [432](#)  
numéros de ligne [48](#)  
NValue [427](#)

## O

ODBCclose [334](#)  
ODBCOpen [334](#)  
ODBCOPENEx [334](#)  
ODBCOutput [335](#)  
onglet Planning  
Editeur TurboIntegrator [100](#)  
onglets  
Editeur TurboIntegrator [78](#)  
OnMinorErrorDoItemSkip [428](#)  
opérateurs arithmétiques [103](#)  
OPTGET [174](#)  
optimisation des cubes [20](#)  
options  
attributs [8](#)  
Cube Viewer, menu [23](#)  
Propriétés d'élément de dimension [30](#)  
options de processus, boîte de dialogue [43](#)  
OPTSET [174](#)  
ouvrir un sous-ensemble, boîte de dialogue [38](#)  
ouvrir une vue, boîte de dialogue [39](#)

## P

paramètres  
Editeur TurboIntegrator [98](#)  
paramètres de connexion  
TM1 Options [74](#)  
paramètres de plage  
Extraction de vue [100](#)  
ParseDate [305](#)  
PAYMT [146](#)  
Préférences [49](#)  
ProcessAttrDelete [262](#)  
ProcessAttrInsert [263](#)  
ProcessAttrN [263](#)  
ProcessAttrNL [264](#)  
ProcessAttrPutN [265](#)  
ProcessAttrPutS [266](#)  
ProcessAttrS [267](#)  
ProcessAttrSL [267](#)  
ProcessBreak [341](#)  
ProcessError [341](#)  
ProcessExists [341](#)  
ProcessQuit [342](#)  
ProcessReturnCode [431](#)  
ProcessRollback [342](#)  
processus  
bouton d'action [2](#)  
Explorateur de serveur [66](#)

processus (*suite*)  
  fonctions de contrôle TurboIntegrator [336](#)  
PrologMinorErrorCount [432](#)  
prologue  
  Editeur TurboIntegrator [98](#)  
propriétés  
  Editeur de dimension, sous-fenêtre [25](#)  
  éditeur de sous-ensemble, sous-fenêtre [68](#)  
  Élément de dimension, sous-fenêtre [30](#)  
  paramètres régionaux [8](#)  
propriétés de paramètres régionaux [8](#)  
PublishSubset [175](#)  
PublishView [176](#), [396](#)  
PV [146](#)

## Q

QUDEFINE [176](#)  
QUDEFINEEX [178](#)  
QUEXPOR [180](#)  
QULOO [181](#)  
QUSUBSET [181](#)

## R

R\_SAVE [182](#)  
RAND [154](#)  
rapports dynamiques  
  TM1INFO [208](#)  
Rapports dynamiques  
  TM1ELLIST [205](#)  
  TM1GLOBALSANDBOX [208](#)  
  TTM1PRIMARYDBNAME [209](#)  
Rechercher [47](#)  
RefreshMdxHierarchy, fonction [375](#)  
régions définies par l'utilisateur [48](#)  
règle  
  fonctions [103](#)  
  fonctions de gestion TurboIntegrator [345](#)  
  fonctions macro [167](#)  
  informations de l'Editeur de sous-ensemble [45](#)  
RemoveCAMIDAssociation [365](#)  
RemoveCAMIDAssociationFromGroup [366](#)  
RemoveClientFromGroup [366](#)  
répliquer  
  Explorateur de serveur [65](#)  
répliquer cube  
  boîte de dialogue [44](#)  
  Explorateur de serveur [66](#)  
requête du journal des transactions  
  boîte de dialogue [75](#)  
  résultats, boîte de dialogue [76](#)  
réservation  
  affectations de sécurité [53](#)  
retour à la ligne [48](#)  
ROUND [155](#)  
RuleLoadFromFile [347](#)  
RunProcess [342](#)

## S

saisie semi-automatique [48](#)  
SAPCharacteristicTexts [426](#)

SaveDataAll [376](#)  
SCAN [160](#)  
sécurité  
  affectations, boîte de dialogue [51](#)  
  fonctions TurboIntegrator [358](#)  
  menu Clients/Groupes [16](#)  
SecurityOverlayCreateGlobalDefault [370](#)  
SecurityOverlayDestroyGlobalDefault [370](#)  
SecurityOverlayGlobalLockCell [369](#)  
SecurityOverlayGlobalLockNode [371](#)  
SecurityRefresh [372](#)  
sélectionner dimension  
  affectations de sécurité [55](#)  
  boîte de dialogue [55](#)  
sélectionner élément  
  boîte de dialogue [55](#)  
  extraction de vue [100](#)  
sélectionner un cube  
  boîte de dialogue [55](#)  
  pour les règles, boîte de dialogue [55](#)  
ServerActiveSandboxGet [349](#)  
ServerActiveSandboxSet [349](#)  
ServerSandboxesDelete [351](#)  
ServerSandboxExists [356](#)  
ServerSandboxGet [356](#)  
ServerSandboxListCountGets [357](#)  
ServerShutdown [377](#)  
serveur  
  Explorateur (Fenêtre principale) [55](#)  
  Explorateur de serveur [56](#)  
  fonctions de manipulation TurboIntegrator [372](#)  
Serveur de connexion [36](#)  
serveur local  
  TM1 Options [74](#)  
SetChoreVerboseMessages [281](#)  
SetDimensionGroupsSecurity [368](#)  
SetElementGroupsSecurity [368](#)  
SetHierarchyElementGroupsSecurity [367](#)  
SetHierarchyGroupsSecurity [367](#)  
SetInputCharacterSet [220](#)  
SetODBCUnicodeInterface [336](#)  
SetOutputEscapeDoubleQuote [223](#)  
SetUseActiveSandboxProperty [358](#)  
SIGN [156](#)  
signets [47](#)  
SIN [156](#)  
sous-ensemble  
  éditeur [68](#)  
  Editeur de sous-ensemble, menu [69](#)  
  Explorateur de serveur [64](#)  
  fonctions de manipulation TurboIntegrator [377](#)  
sous-fenêtre de droite (Propriétés)  
  Explorateur de serveur [55](#)  
sous-fenêtre de gauche (arborescence)  
  Explorateur de serveur [55](#)  
SQRT [156](#)  
STET [149](#), [217](#)  
STR [161](#)  
StringGlobalVariable(NomVariable) [430](#)  
StringSessionVariable(NomVariable) [432](#)  
StringToNumber [224](#)  
StringToNumberEx [224](#)  
SUBDEFINE [182](#)  
SUBNM [203](#)

[SUBPICK 183](#)  
[SubsetAliasSet 385](#)  
[SubsetAttrDelete 274](#)  
[SubsetAttrInsert 273](#)  
[SubsetATTRN 269](#)  
[SubsetATTRNL 270](#)  
[SubsetAttrPutN 272](#)  
[SubsetAttrPutS 271](#)  
[SubsetATTRS 268](#)  
[SubsetATTRSL 269](#)  
[SubsetCreate 386](#)  
[SubsetCreateByMDX 387](#)  
[SubsetDeleteAllElements 388](#)  
[SubsetDestroy 389](#)  
[SubsetElementDelete 389](#)  
[SubsetElementExists 390](#)  
[SubsetElementGetIndex 390](#)  
[SubsetElementInsert 391](#)  
[SubsetExists 391](#)  
[SubsetExpandAboveSet 392](#)  
[SubsetFormatStyleSet 393](#)  
[SubsetGetElementName 393](#)  
[SubsetGetSize 394](#)  
[SubsetIsAllSet 394](#)  
[SubsetMDXGet 394](#)  
[SubsetMDXSet 395](#)  
[SUBSIZ 204](#)  
[SUBST 163](#)  
[supprimer le retrait 47](#)  
[SValue 428](#)  
[synchronized\(\) 343](#)  
[syntaxe 343](#)

## T

[T\\_CLEAR 184](#)  
[T\\_CREATE 184](#)  
[T\\_CREATE16 185](#)  
[T\\_PICK 186](#)  
[T\\_SAVE 186](#)  
[TABDIM 131, 205](#)  
 tâche
 

- [Assistant Définition 14](#)
- [Explorateur de serveur 67](#)
- [fonctions TurboIntegrator de gestion 280](#)
- [quitter 281](#)

[TAN 157](#)  
[TextOutput 225](#)  
[TIME 123](#)  
[TIMST 123](#)  
[TIMVL 125](#)  
[TM1ELLIST 205](#)  
[TM1GLOBSANDBOX 208](#)  
[TM1INFO 208](#)  
[Tm1p.xla 167](#)  
[TM1PRIMARYDBNAME 209](#)  
[TM1ProcessError.log 428](#)  
[TM1RECALC 186](#)  
[TM1RECALC1 187](#)  
[TM1RptEIsConsolidated 209](#)  
[TM1RPTTELISCONSOLIDATED 214](#)  
[TM1RptEIsExpanded 210](#)  
[TM1RptElLev 210](#)  
[TM1RPTELLSEXPANDED 214](#)

[TM1RptFilter 210](#)  
[TM1RptRow 211](#)  
[TM1RptTitle 213](#)  
[TM1RptView 214](#)  
[TM1User 214, 416](#)  
[TODAY 127](#)  
 tous les écrans
 

- [Assistant d'impression de rapports 39](#)

[TRIM 163](#)  
 TurboIntegrator
 

- [fonctions 217](#)
- [limites 217](#)
- [variables globales 429](#)
- [variables utilisateur 432](#)

## U

[UNDEFINEDCELLVALUE 119](#)  
[UPPER 164](#)  
[UTF-8 220](#)

## V

[Value\\_Is\\_String 429](#)  
 variables
 

- [global 429](#)
- [globales implicites 430](#)
- [onglet Editeur TurboIntegrator 92](#)
- [utilisateur TurboIntegrator 432](#)

 variables globales [429](#)  
 variables globales implicites [430](#)  
 variables locales [419](#)  
[VC 145](#)  
[vérifier la syntaxe 46](#)  
 verrou
 

- [affectations de sécurité 53](#)

[VIEW 215](#)  
[ViewAttrDelete 274](#)  
[ViewAttrInsert 275](#)  
[ViewAttrN 275](#)  
[ViewAttrNL 276](#)  
[ViewAttrPutN 277](#)  
[ViewAttrPutS 278](#)  
[ViewAttrS 279](#)  
[ViewAttrSL 279](#)  
[ViewColumnDimensionSet 398](#)  
[ViewColumnSuppressZeroesSet 398](#)  
[ViewConstruct 399](#)  
[ViewCreate 399](#)  
[ViewCreateByMDX 400](#)  
[ViewDestroy 401](#)  
[ViewExists 402](#)  
[ViewExtractFilterByTitlesSet 402](#)  
[ViewExtractSkipCalcsSet 403](#)  
[ViewExtractSkipConsolidatedStringsSet 404](#)  
[ViewExtractSkipRuleValuesSet 405](#)  
[ViewExtractSkipZeroesSet 405](#)  
[ViewMDXGet 407](#)  
[ViewMDXSet 406](#)  
[ViewRowDimensionSet 407](#)  
[ViewRowSuppressZeroesSet 408](#)  
[ViewSubsetAssign 409](#)  
[ViewSuppressZeroesSet 409](#)

[ViewTitleDimensionSet 410](#)  
[ViewTitleElementSet 410](#)  
[ViewZeroOut 411](#)  
vue  
    fenêtre Extraction [100](#)  
    fonctions de manipulation TurboIntegrator [395](#)  
    styles, boîte de dialogue [101](#)  
vue de cube  
    Explorateur de serveur [64](#)  
[VUSLICE 187](#)

## W

[W\\_DBSENABLE 187](#)  
[While 344](#)  
[WildcardFileSearch 416](#)

## Y

[YEAR 128](#)



