

IBM Cognos Transformer  
Version 10.2.2

*Guide d'automatisation*



**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 515.

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 CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.ibm.com/ca/fr> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
Direction Qualité  
17, avenue de l'Europe  
92275 Bois-Colombes Cedex*

**Informations sur le produit**

Le présent document s'applique à IBM Cognos Business Intelligence version 10.2.2 et peut aussi s'appliquer aux éditions ultérieures de ce produit.

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

© Copyright IBM Corporation 2007, 2014.

---

# Table des matières

<b>Avis aux lecteurs canadiens</b> . . . . .	<b>xiii</b>
<b>Introduction</b> . . . . .	<b>xv</b>
<b>Chapitre 1. Nouveautés</b> . . . . .	<b>1</b>
Nouveautés de la version 10.2.2 . . . . .	1
Nouveautés de la version 10.2.1 . . . . .	1
Nouveautés de la version 10.2.0 . . . . .	1
<b>Chapitre 2. Présentation</b> . . . . .	<b>3</b>
Présentation de la création d'objets . . . . .	3
Présentation de la bibliothèque de types pour Visual Basic . . . . .	4
Présentation de la bibliothèque de types pour Visual C++ et Visual Studio 2009 C#. . . . .	5
Présentation de la bibliothèque de types de Transformer. . . . .	6
Présentation des méthodes. . . . .	7
Présentation des listes de valeurs et des constantes . . . . .	7
Grille hiérarchique des objets de Transformer . . . . .	8
Grille hiérarchique A des objets de Transformer . . . . .	10
Grille hiérarchique B des objets de Transformer . . . . .	11
Grille hiérarchique C des objets de Transformer . . . . .	12
Grille hiérarchique D des objets de Transformer . . . . .	12
Grille hiérarchique E des objets de Transformer . . . . .	13
Paramètres préférentiels du kit de développement de logiciels . . . . .	13
Modèles IBM Cognos Series 7 . . . . .	13
<b>Chapitre 3. Collections</b> . . . . .	<b>15</b>
Collection Associations . . . . .	16
Collection CalculationDefinitions . . . . .	17
Collection Categories . . . . .	18
Collection CategorySets . . . . .	19
Collection ChildCubes . . . . .	19
Collection Columns. . . . .	20
Collection CubeCustomViews . . . . .	21
Collection Cubes . . . . .	21
Collection CurrencyRates . . . . .	22
Collection CurrencyRecords . . . . .	22
Collection CurrencyTables . . . . .	23
collection CustomViews . . . . .	24
Collection DataSources . . . . .	25
Collection DimensionLevels . . . . .	26
Collection Dimensions. . . . .	26
Collection DrillDowns . . . . .	27
Collection DrillThroughTargets . . . . .	28
Collection Filters . . . . .	29
collection LevelCategories . . . . .	30
Collection LevelDrillDowns . . . . .	30
Collection Levels . . . . .	31
Collection Measures . . . . .	32
Collection Names . . . . .	33
Collection Namespaces . . . . .	33
Collection PackageDatasourceConnections . . . . .	34
Collection Packages. . . . .	35
Collection Prompts . . . . .	36
Collection Queries . . . . .	37

Collection Reports . . . . .	38
Collection SecurityObjects . . . . .	38
Collection Signons . . . . .	39
Collection SuspendedModels . . . . .	40
Collection Views. . . . .	41

**Chapitre 4. Objets . . . . . 43**

Objet Application . . . . .	45
Objet Association . . . . .	48
Objet CalculationDefinition . . . . .	49
Objet Category . . . . .	50
Objet CategorySet . . . . .	53
Objet ChildCube. . . . .	55
Objet Column . . . . .	56
Objet CrossTabDataSource . . . . .	59
Objet Cube . . . . .	61
Objet CubeGroup . . . . .	64
Objet CurrencyRate. . . . .	67
Objet CurrencyRecord . . . . .	68
Objet CurrencyTable . . . . .	70
Objet CustomView . . . . .	72
Objet DataSource . . . . .	73
Objet DateDimension . . . . .	75
Objet DateDrillDown . . . . .	78
Objet DateLevel . . . . .	80
Objet DateWizard . . . . .	84
Objet DbDataSource . . . . .	85
Objet Dimension. . . . .	88
Objet DrillDown. . . . .	90
Objet DrillThroughTarget . . . . .	91
Objet Filter . . . . .	93
Objet FlatFileDataSource . . . . .	93
Objet IqdDataSource . . . . .	95
Objet de niveau . . . . .	98
Objet Measure . . . . .	100
Objet Model. . . . .	104
Objet Name . . . . .	107
Objet Namespace . . . . .	107
Objet Package . . . . .	108
Objet PackageDatasourceConnection. . . . .	109
Objet Prompt . . . . .	110
Objet Query . . . . .	111
Objet Report. . . . .	113
Objet SecurityObject . . . . .	114
Objet de code d'accès. . . . .	115
Objet SpecialCategory . . . . .	117
Objet SuspendedModel . . . . .	119
Objet View . . . . .	120

**Chapitre 5. Méthodes . . . . . 123**

Méthode Add () . . . . .	127
Méthode Add (Categories) . . . . .	128
Méthode Add (CustomViews) . . . . .	129
Méthode Add (DrillThroughTargets). . . . .	129
Méthode Add (Objects) . . . . .	130
Méthode AddDeployLocation . . . . .	131
Méthode AddToCustomView . . . . .	132
Méthode AddToFolder . . . . .	133
Méthode AssociateWith . . . . .	133
Méthode CheckLocalPowerCubes . . . . .	134

Méthode CheckModel . . . . .	135
Méthode CleanHouse. . . . .	136
Méthode ClearDeployLocations . . . . .	137
Méthode Close . . . . .	137
Méthode ConnectWithCategory . . . . .	138
Méthode CreateAlternateDrillDown . . . . .	139
Méthode CreateDateDimension . . . . .	139
Méthode CreateMDCFile . . . . .	140
Méthode CreateMDCFile . . . . .	141
Méthode Delete . . . . .	141
Méthode DeleteAllCustomViews . . . . .	143
Méthode DeleteAllSecurityObjects . . . . .	144
Méthode DeployCube . . . . .	144
Méthode DeployCubes . . . . .	145
Méthode DimensionAssociateWith . . . . .	145
Méthode DoAutoDesign. . . . .	146
Méthode FindCategoryByCatCode . . . . .	147
Méthode GenerateCategories . . . . .	147
Méthode GenerateDateCategories . . . . .	148
Méthode GetDefaultCategory . . . . .	149
Méthode GetViewStatus . . . . .	149
Méthode isExcludeDateDimension . . . . .	150
Méthode isExcludeDateLevel . . . . .	150
Méthode isExcludeDimension . . . . .	151
Méthode isExcludeLevel. . . . .	152
Méthode Item . . . . .	152
Méthode Item () . . . . .	154
Méthode Item () . . . . .	156
Méthode LoadCurrencyTable . . . . .	156
Méthode Logoff . . . . .	157
Méthode Logon . . . . .	157
Méthode Move . . . . .	158
Méthode MoveToCategory . . . . .	159
Méthode MoveToLevel . . . . .	160
Méthode NewModel . . . . .	161
Méthode OpenModel. . . . .	162
Méthode OpenSuspendedModel . . . . .	163
Méthode PublishDatasource . . . . .	163
Méthode PublishPackage . . . . .	164
Méthode Remove . . . . .	165
Méthode RemoveCubeCustomView . . . . .	167
Méthode RemoveFromFolder . . . . .	167
Méthode RemoveSuspendedModel . . . . .	168
Méthode ResetPartitions. . . . .	169
Méthode Save . . . . .	169
Méthode SaveAs . . . . .	169
Méthode SetAllocation . . . . .	170
Méthode SetDefaultCategory . . . . .	171
Méthode SetDeployType. . . . .	172
Méthode SetExcludeDateDimension . . . . .	173
Méthode SetExcludeDateLevel. . . . .	174
Méthode SetExcludeDimension . . . . .	175
Méthode SetExcludeLevel . . . . .	175
Méthode SetViewStatus . . . . .	176
Méthode TestBuild . . . . .	177
Méthode Update . . . . .	178
Méthode Verify. . . . .	180

<b>Chapitre 6. Propriétés. . . . .</b>	<b>181</b>
Propriété ActivityMeasure . . . . .	199
Propriété Aggregate . . . . .	199

Propriété AllocationMeasure . . . . .	200
Propriété AllocationType . . . . .	201
Propriété AllowCurrencyConversion. . . . .	202
Propriété AllowDrillThrough . . . . .	203
Propriété AlternateQueryPath . . . . .	204
Propriété AltMDCFile . . . . .	205
Propriété AlwaysUseTransformerSignon . . . . .	205
Propriété Apex . . . . .	206
Propriété Application. . . . .	207
Propriété AssociationRole . . . . .	210
Propriété Associations . . . . .	210
Propriété AssociationType . . . . .	211
Propriété AutoLogon . . . . .	212
Propriété AutoSummary . . . . .	213
Propriété BlankSubstitute . . . . .	213
Propriété BlockParentTotals. . . . .	214
Propriété CacheCrossTabs . . . . .	215
Propriété CalculationDefinitions . . . . .	215
Propriété CAMID . . . . .	216
Propriété CanAllocate . . . . .	216
Propriété CanAllocateByMeasure . . . . .	217
Propriété CanAllocateMeasure. . . . .	219
Propriété Categories . . . . .	220
Propriété Category . . . . .	221
Propriété CategoryCount . . . . .	221
Propriété CategoryCountLevel. . . . .	222
Propriété CategorySets . . . . .	223
Propriété CharacterType . . . . .	223
Propriété ChildCategories . . . . .	224
Propriété ChildCubes. . . . .	225
Propriété ChildCustomViews . . . . .	225
Propriété ChildMeasures . . . . .	226
Propriété Code . . . . .	227
Propriété Columns . . . . .	227
Propriété ColumnsLoaded . . . . .	228
Propriété CompressMDC . . . . .	229
Propriété Connection . . . . .	229
Propriété Consolidate. . . . .	230
Propriété Context . . . . .	231
Propriété ContextLevel . . . . .	232
Propriété ContextOffset . . . . .	233
Propriété ConvergenceLevel . . . . .	233
Propriété Count . . . . .	234
Propriété CountryCode . . . . .	236
Propriété CubeCodePage . . . . .	237
Propriété CubeCreation . . . . .	238
Propriété CubeCustomViews . . . . .	239
Propriété Cubes . . . . .	240
Propriété CubeStamp . . . . .	240
Propriété CurrencyCountryLabel . . . . .	241
Propriété CurrencyDecimals . . . . .	241
Propriété CurrencyFormatOverride . . . . .	242
Propriété CurrenciesEMU . . . . .	243
Propriété CurrenciesEuro . . . . .	244
Propriété CurrencyRates. . . . .	244
Propriété CurrencyRecord . . . . .	245
Propriété CurrencyRecords . . . . .	246
Propriété CurrencySymbol . . . . .	246
Propriété CurrencyTable . . . . .	247
Propriété CurrencyTables . . . . .	248
Propriété CurrencyTableType . . . . .	248

Propriété CurrentModel . . . . .	249
Property CurrentValueIndex . . . . .	250
Propriété CustomView . . . . .	250
Propriété CustomViews . . . . .	251
Propriété DataCharacterSet . . . . .	251
Propriété DataClass . . . . .	252
Propriété DataRange . . . . .	253
Propriété DataSource . . . . .	253
Propriété DataSourcePath . . . . .	254
Propriété DataSources . . . . .	254
Propriété DataSourceWindowsLocation . . . . .	255
Propriété DataTemporaryFilePath . . . . .	256
Propriété DateDegreeofDetail . . . . .	256
Propriété DateDegreeofDetailLevelName . . . . .	257
Propriété DateFormat . . . . .	258
Propriété DateFunction . . . . .	259
Propriété DateInputFormat . . . . .	260
Propriété DateLevel . . . . .	260
Propriété DateWizard . . . . .	261
Propriété DecimalPoint . . . . .	262
Propriété Decimals . . . . .	262
Propriété DefaultCategoryOrderBy . . . . .	263
Propriété DefaultDateFormat . . . . .	263
Propriété Description . . . . .	264
Propriété DesiredPartitionSize . . . . .	265
Propriété DetachDataSource . . . . .	266
Propriété DetailLevel . . . . .	267
Propriété Dimension . . . . .	268
Propriété DimensionInclude . . . . .	268
Propriété DimensionLevels . . . . .	269
Propriété DimensionName . . . . .	270
Propriété Dimensions . . . . .	271
Propriété DimensionView . . . . .	271
Propriété DimensionViewType . . . . .	273
Propriété DisplayName . . . . .	274
Propriété DrillCode . . . . .	274
Propriété DrillDowns . . . . .	275
Propriété DrillInclusion . . . . .	275
Propriété DrillThroughTargets . . . . .	276
Propriété DuplicateRollup . . . . .	277
Propriété DuplicateWeight . . . . .	278
Propriété EarliestDate . . . . .	279
Propriété EMUEntryDate . . . . .	279
Propriété EnableMessageLogging . . . . .	280
Propriété EnableTimePeriod . . . . .	281
Propriété EstimatedRows . . . . .	281
Propriété ExcludeAutoPartition . . . . .	282
Propriété ExpressionText . . . . .	283
Propriété External . . . . .	284
Propriété FieldSeparator . . . . .	285
Propriété FileName . . . . .	286
Propriété Filters . . . . .	286
Propriété FindCategoryByCatCode . . . . .	287
Propriété Format . . . . .	288
Propriété FormatDecimals . . . . .	289
Propriété FullName . . . . .	290
Propriété GenerateCategories . . . . .	290
Propriété GenerateDateCategories . . . . .	291
Propriété GenerateDates . . . . .	292
Propriété GeneratePowerCube . . . . .	293
Propriété GenerateTimePeriod . . . . .	293

Propriété Group . . . . .	294
Propriété GroupDimension . . . . .	295
Propriété GroupLevel. . . . .	296
Propriété HasSubdimension . . . . .	297
Propriété HideValue . . . . .	297
Propriété ID. . . . .	298
Propriété IgnoreMissingValue . . . . .	298
Propriété Inclusion . . . . .	299
Propriété IncrementalUpdate . . . . .	300
Propriété InputScale . . . . .	301
Propriété IsAnyColumnMismatched . . . . .	302
Propriété IsBad . . . . .	303
Propriété IsExpressionValid. . . . .	303
Propriété IsFolder . . . . .	304
Propriété IsManual . . . . .	305
IPropriété sMDCInUse . . . . .	305
Propriété IsolationLevel . . . . .	306
Propriété IsPrimary . . . . .	307
Propriété IsTimeBasedPartitionedCube . . . . .	308
Propriété KeyName . . . . .	309
Propriété Label . . . . .	309
Propriété LastUseDate . . . . .	310
Propriété LatestDate . . . . .	311
Propriété Level . . . . .	312
Propriété LevelCategories . . . . .	312
Propriété LevelDrillDowns . . . . .	313
Propriété Levels . . . . .	314
Propriété LocalPath . . . . .	314
Propriété LogErrorLevel. . . . .	315
Propriété LogFileAppend . . . . .	316
Propriété LogFileName . . . . .	317
Propriété LogFilesPath . . . . .	317
Propriété Lunar . . . . .	318
Propriété ManualCurrentPeriod . . . . .	318
Propriété MaximizeSpeed . . . . .	319
Propriété MaxNumPartLevels . . . . .	320
Propriété MaxTransactionNumber . . . . .	321
Propriété MDCFile . . . . .	322
Propriété MeasureInclude . . . . .	322
Propriété MeasureName. . . . .	323
Propriété Measures . . . . .	324
Propriété MeasureType . . . . .	324
Propriété MissingValue . . . . .	325
Propriété ModelName . . . . .	326
Propriété ModelsPath. . . . .	326
Propriété ModelTemporaryFilesPath . . . . .	327
Propriété ModelType . . . . .	328
Propriété MonthType . . . . .	328
Propriété Name . . . . .	329
Propriété Namespaces . . . . .	331
Propriété NewCatsLocked . . . . .	331
Propriété ObjectCAMID . . . . .	332
Propriété ObjectName . . . . .	333
Propriété Optimize . . . . .	333
Propriété OrderByDescending . . . . .	334
Propriété OrderByStorageType. . . . .	335
Propriété Origin . . . . .	336
Propriété OriginalName . . . . .	336
Propriété Orphanage . . . . .	337
Propriété OutputScale . . . . .	338
Propriété Packages . . . . .	339



Propriété PackagesDataSourceConnections . . . . .	340
Propriété Parent . . . . .	340
Propriété ParentCategories . . . . .	343
Propriété Partition . . . . .	344
Propriété Password . . . . .	345
Propriété PatFile . . . . .	346
Propriété Path . . . . .	347
Propriété PopulateByDataSource . . . . .	348
Propriété Position . . . . .	348
Propriété PowerCubesPath . . . . .	349
Propriété PowerPlayPath . . . . .	349
Propriété Precision . . . . .	350
Propriété PromptForPassword . . . . .	351
Propriété Prompts . . . . .	351
Propriété PromptValueType . . . . .	352
Propriété QualifiedName . . . . .	352
Propriété QuarterType . . . . .	353
Propriété Queries . . . . .	354
Propriété QyPath . . . . .	354
Propriété Rate . . . . .	355
Propriété RefName . . . . .	356
Propriété RefreshDescription . . . . .	356
Propriété RefreshLabel . . . . .	357
Propriété RefreshShortName . . . . .	358
Propriété RegularRollup . . . . .	358
Propriété RegularWeight . . . . .	359
Propriété Reports . . . . .	360
Propriété ReverseSign . . . . .	361
Propriété Rollup . . . . .	362
Propriété RollupTiming . . . . .	362
Propriété RowsAsSample . . . . .	363
Propriété RowsChecked . . . . .	364
Propriété RunningPeriods . . . . .	364
Propriété SecurityObjects . . . . .	365
Propriété Server . . . . .	366
Propriété ServerModelPath . . . . .	367
Propriété ServerPath . . . . .	367
Propriété ServerQuery . . . . .	368
Propriété ServicesBuildNumber . . . . .	368
Propriété ServicesVersionText . . . . .	369
Propriété SetsCurrentPeriod . . . . .	369
Propriété ShortName . . . . .	370
Propriété Signon . . . . .	371
Propriété SignOnNamespace . . . . .	372
Propriété Signons . . . . .	373
Propriété SignonType . . . . .	373
Propriété Size . . . . .	374
Propriété SortComparisonRule . . . . .	374
Propriété SourceType . . . . .	375
Propriété SpecialCategoryCount . . . . .	376
Propriété SQLExpression . . . . .	377
Propriété Status . . . . .	377
Propriété StorageType . . . . .	378
Propriété StreamExtractAllowed . . . . .	379
Propriété StreamExtractSize . . . . .	379
Propriété SummaryLevel . . . . .	380
Propriété SuppressNull . . . . .	380
Propriété SuspendedModels . . . . .	381
Propriété TargetLevel . . . . .	382
Propriété TargetOffset . . . . .	383
Propriété ThousandPoint . . . . .	383

Propriété Time . . . . .	384
Propriété TimeArrayColumn . . . . .	384
Propriété TimeArrayStartMonth . . . . .	385
Propriété TimeArrayType . . . . .	386
Propriété TimeRank . . . . .	387
Propriété TimeStamp . . . . .	387
Propriété TimeStateRollup . . . . .	388
Propriété TimeStateWeight . . . . .	389
Propriété ToDateLevel . . . . .	389
Propriété TransdaPath . . . . .	390
Propriété TransformerSignon . . . . .	391
Propriété Type . . . . .	391
Propriété Unique . . . . .	393
Propriété UniqueMove . . . . .	395
Propriété UseAltMDCFFile . . . . .	395
Propriété User . . . . .	396
Propriété UserCAMID . . . . .	397
Propriété UserID . . . . .	397
Propriété Value . . . . .	398
Propriété ValuesCount . . . . .	398
Propriété Version . . . . .	399
Propriété Views . . . . .	400
Propriété ViewType . . . . .	400
Propriété WeekAdd . . . . .	401
Propriété WeekSpan . . . . .	402
Propriété WeekStart . . . . .	403
Propriété WeekStartDay . . . . .	403
Propriété WorkingDay . . . . .	404
Propriété WorkingDays . . . . .	405
Propriété YearStartDay . . . . .	407
Propriété YearType . . . . .	407

**Chapitre 7. Constantes . . . . . 409**

Liste des valeurs de xtrAllocationType . . . . .	409
Liste des valeurs de xtrAssociationRole. . . . .	410
Liste des valeurs de xtrAssociationType . . . . .	411
Liste des valeurs de xtrCharacterType . . . . .	411
Liste des valeurs de xtrCubeConsolidate . . . . .	412
Liste des valeurs de xtrCubeCreation . . . . .	412
Liste des valeurs de xtrCubeOptimize . . . . .	413
Liste des valeurs de xtrCubeStatus . . . . .	414
Liste des valeurs de xtrCurrencyTableType . . . . .	415
Liste des valeurs de xtrDataClass. . . . .	415
Liste des valeurs de xtrDateCategoriesGeneration . . . . .	416
Liste des valeurs de xtrDateFormat . . . . .	416
Liste des valeurs de xtrDateLevel. . . . .	417
liste des valeurs de xtrDeployType . . . . .	418
Liste des valeurs de xtrDuplicateRollup . . . . .	418
Liste des valeurs de xtrGenerateOptions . . . . .	419
Liste des valeurs de xtrInclusion . . . . .	420
Liste des valeurs de xtrMeasureType . . . . .	421
Liste des valeurs de xtrMissingValue . . . . .	421
Liste des valeurs de xtrObjectType . . . . .	422
Liste des valeurs de xtrOrigin . . . . .	423
Liste des valeurs de xtrPowerCubeGeneration . . . . .	424
Liste des valeurs de xtrPreferences . . . . .	425
liste des valeurs de xtrPromptValueType . . . . .	426
Liste des valeurs de xtrRollup . . . . .	427
Liste des valeurs de xtrRollupTiming . . . . .	428
Liste des valeurs xtrSecurityType. . . . .	429
Liste des valeurs de xtrSourceType . . . . .	429

Liste des valeurs de xtrSpecialFunction . . . . .	431
Liste des valeurs de xtrStorage . . . . .	432
Liste des valeurs de xtrTimeAggregate . . . . .	433
Liste des valeurs de xtrTimeArrayType . . . . .	434
Liste des valeurs de xtrTimeRollup . . . . .	434
Liste des valeurs de xtrTimeType . . . . .	436
Liste des valeurs de xtrViewStatus . . . . .	437
Liste des valeurs de xtrViewType . . . . .	438
Liste des valeurs de xtrWeekAdd . . . . .	438
Liste des valeurs de xtrWeekDay . . . . .	439
Liste des valeurs de xtrWeekspan . . . . .	440

**Chapitre 8. Equivalents IU . . . . . 441**

Collections . . . . .	441
Objets . . . . .	443
Méthodes. . . . .	445
Propriétés . . . . .	447
Listes des valeurs et constantes . . . . .	470

**Chapitre 9. Exemples . . . . . 475**

Exemple d'ouverture d'un modèle et de spécification d'un ordre par association . . . . .	475
Exemple d'ouverture d'un modèle et d'ajout d'un calcul . . . . .	476
Exemple de création d'une catégorie de date relative . . . . .	477
Exemple d'ajout d'un groupe de cubes . . . . .	478
Exemple d'ajout d'une source de données supplémentaire à un modèle . . . . .	479
Exemple d'ouverture d'un modèle et de modification des propriétés du cube . . . . .	480
Exemple de création d'une vue personnalisée. . . . .	481
Exemple d'ouverture d'un modèle et d'ajout d'un enregistrement de devise. . . . .	482
Exemple de création d'un cube à l'aide des méthodes DoAutoDesign et TestBuild . . . . .	483
Exemple de sélection, de modification et de mise à jour d'une dimension . . . . .	484
Exemple de suppression d'un niveau à partir d'une collection de niveaux . . . . .	485
Exemple de déplacement d'un objet de mesure et de modification du cumul de la mesure Revenus . . . . .	485
Exemple de création d'une partition et de vérification du modèle . . . . .	486
Exemple d'ouverture d'un modèle et d'accès au détail du PowerCube . . . . .	487
Exemple d'ajout du rôle Auteurs à une vue personnalisée . . . . .	487
Exemple de recherche d'un modèle en attente . . . . .	489
Exemple d'ouverture d'un modèle et de création d'une vue des dimensions . . . . .	489
Exemple d'ajout d'un groupe de cubes à un modèle . . . . .	490
Exemple de création d'un modèle et de mise à jour des propriétés pour une dimension de date. . . . .	491
Exemple de création d'une branche de niveaux secondaire . . . . .	492
Exemple d'utilisation de l'objet DateWizard pour créer une dimension de temps . . . . .	494
Exemple de création d'une mesure de nombre de catégories et d'ajout à un modèle . . . . .	495
Exemple d'ajout d'une vue des dimensions à un modèle . . . . .	496
Exemple de déplacement d'une catégorie enfant vers un autre parent . . . . .	496
Exemple d'ajout d'une table dans un fichier et de chargement des données. . . . .	497
Exemple de génération d'une dimension de temps basée sur une année lunaire . . . . .	499
Exemple de déplacement d'une catégorie enfant vers une autre catégorie parent . . . . .	500
Exemple de définition d'attributs pour une application . . . . .	501
Exemple d'ajout et de suppression d'un pack . . . . .	502
Exemple d'ajout et de suppression d'un rapport . . . . .	503
Exemple de création d'une requête . . . . .	504
Exemple de création et de suppression des filtres d'un modèle . . . . .	505
Exemple de création d'une invite à valeur unique . . . . .	506
Exemple de création d'une invite à valeurs multiples . . . . .	508
Exemple de création d'un modèle et de publication d'un PowerCube. . . . .	509
Exemple de copie et d'activation d'un PowerCube . . . . .	510
Exemple de création d'un modèle avec un code d'accès et une source de données IQD . . . . .	511
Exemple de création d'un modèle avec un code d'accès et une source de données de pack . . . . .	512

**Remarques . . . . . 515**



---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

### Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

## Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

## Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

---

## Introduction

Ce document est destiné à être utilisé avec IBM® Cognos Transformer, le composant de modélisation OLAP fourni avec IBM Cognos Business Intelligence.

Vous pouvez utiliser ce *Guide d'automatisation IBM Cognos Transformer* comme référence lors de l'écriture de scripts d'automatisation pour IBM Cognos Transformer.

### Audience

Ce manuel s'adresse aux utilisateurs expérimentés de Transformer qui en connaissent bien les concepts, les fonctionnalités et la terminologie. Seules les fonctions d'automatisation OLE de Transformer sont documentées. Les fonctions de base Transformer ne sont pas expliquées.

### Recherche d'informations

Pour rechercher la documentation des produits sur le Web, y compris toutes les documentations traduites, accédez à la page IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

### Fonctions d'accessibilité

Ce produit ne prend pas en charge les fonctions d'accessibilité permettant aux utilisateurs souffrant d'un handicap physique, comme une mobilité réduite ou une vision limitée, d'utiliser le 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.

### Clause de décharge relative aux exemples

La société Vacances et aventure, Ventes VA, toute variation du nom Vacances et aventure, ainsi que les exemples de planification, illustrent des opérations commerciales fictives, avec des exemples de données utilisées pour développer des exemples d'applications, destinées à l'usage d'IBM et de ses clients. Ces données fictives comprennent des exemples de données pour des transactions de ventes, la distribution de produits, la finance et les ressources humaines. Toute ressemblance avec des noms, adresses, numéros de contact ou valeurs de transaction existants est purement fortuite. D'autres exemples de fichiers peuvent contenir des données fictives créées manuellement ou par ordinateur, des données basées sur les faits qui sont compilées à partir de sources universitaires ou publiques, ou encore des données utilisées avec les autorisations du détenteur des droits d'auteur, à utiliser en tant qu'exemples pour développer des exemples d'applications. Les noms de produits référencés peuvent être des marques commerciales de leurs propriétaires

respectifs. Toute duplication effectuée sans autorisation est interdite.



---

## Chapitre 1. Nouveautés

Cette section contient la liste des nouvelles fonctions et des fonctions modifiées qui sont disponibles dans cette édition, et ces informations pour les éditions précédentes. La connaissance de ces informations va vous permettre de planifier vos mises à niveau et vos stratégies de déploiement d'application, ainsi que les critères de formation requis pour vos utilisateurs.

Pour plus d'informations sur la mise à niveau, voir le *Guide d'installation et de configuration d'IBM Cognos Business Intelligence*.

Pour en savoir davantage sur l'ensemble des nouvelles fonctions de cette édition IBM Cognos Business Intelligence, voir le manuel *IBM Cognos Business Intelligence - Guide des nouveautés*.

Pour rechercher la documentation produit sur le Web, y compris tous les documents traduits, accédez à IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

Pour consulter la liste actualisée des environnements pris en charge par les produits IBM Cognos Business Intelligence, y compris des informations sur les systèmes d'exploitation, les correctifs, les navigateurs, les serveurs Web, les serveurs d'annuaire, les serveurs de base de données et les serveurs d'applications, reportez-vous à la page IBM Software Product Compatibility Reports ([www.ibm.com/support/docview.wss?uid=swg27042164](http://www.ibm.com/support/docview.wss?uid=swg27042164)).

---

### Nouveautés de la version 10.2.2

La version 10.2.2 ne contient pas de nouvelles fonctions.

---

### Nouveautés de la version 10.2.1

La version 10.2.1 ne contient pas de nouvelles fonctions.

---

### Nouveautés de la version 10.2.0

La version 10.2.0 ne contient pas de nouvelles fonctions.



---

## Chapitre 2. Présentation

L'automatisation OLE de Transformer utilise une interface en langage de programmation comme alternative à l'interface utilisateur de Transformer. Le présent document comprend des informations relatives aux équivalents IU de certaines des méthodes et propriétés OLE, afin de vous permettre de vous familiariser avec Transformer.

Elle présente un modèle Transformer sous la forme d'un ensemble de collections et d'objets qui sont modifiés par des propriétés et sur lesquels agissent des méthodes. Utilisez-la pour créer et gérer des dimensions, des niveaux, des sources de données, des mesures, des catégories, des branches de niveaux, ainsi que tous les autres objets du modèle, voire pour créer des PowerCubes.

Quand vous créez un modèle dans OLE, vous devez créer vos objets et leur affecter des valeurs dans un ordre hiérarchique. Par exemple, avant de pouvoir créer un niveau, vous devez créer la dimension dans laquelle il figure.

Vous pouvez utiliser la grille hiérarchique «Grille hiérarchique des objets de Transformer», à la page 8 pour déterminer l'ordre dans lequel vous devez créer des objets.

Pour en savoir davantage sur la création de macros dans Transformer, reportez-vous aux rubriques suivantes.

- «Présentation de la création d'objets»
- «Présentation de la bibliothèque de types pour Visual Basic», à la page 4
- «Présentation de la bibliothèque de types pour Visual C++ et Visual Studio 2009 C#», à la page 5
- «Présentation de la bibliothèque de types de Transformer», à la page 6
- «Présentation des méthodes», à la page 7
- «Présentation des listes de valeurs et des constantes», à la page 7

---

### Présentation de la création d'objets

La procédure de création d'objets dans Transformer OLE dépend de l'éditeur employé, du langage utilisé et des résultats que vous souhaitez obtenir.

La plupart des exemples contenus dans la documentation OLE de Transformer créent des objets de manière générique, comme ci-dessous :

```
Dim objTransApp As Object
Dim objModel As Object
Dim objDataSource As Object
Dim objDimension As Object
```

Les objets créés selon cette méthode générique sont compatibles avec VB.NET.

Cette méthode ne vous permet toutefois pas de profiter des fonctionnalités de la bibliothèque de types de Transformer. Pour accéder à cette bibliothèque, vous devez utiliser Microsoft Visual Basic ou un éditeur compatible VB et créer vos objets de la manière suivante :

```
Dim objTransApp As Application
Dim objModel As Model
Dim objDataSource As DataSource
Dim objDimension As Dimension
```

---

## Présentation de la bibliothèque de types pour Visual Basic

Une bibliothèque de types est un fichier binaire qui contient des interfaces de classe et des listes de valeurs (énumérateurs) reconnus par un serveur d'automatisation OLE. Les bibliothèques de types sont prises en charge par les outils de développement, tels que Microsoft Visual Basic, Visual C# et Visual C++. Les bibliothèques de types qui sont intégrées à l'environnement de développement vous permettent d'extraire des informations d'automatisation pendant la compilation et l'exécution d'une application.

La bibliothèque de types de Transformer contient des informations relatives aux objets et aux collections, ainsi qu'à leurs propriétés et méthodes. Elle offre aussi des fonctionnalités telles que :

- la vérification des erreurs à la compilation plutôt qu'à l'exécution,
- la liaison précoce des méthodes et des propriétés plutôt que la liaison tardive, ce qui augmente la vitesse d'exécution,
- des fonctions d'aide avancées lors de la création de scripts.

Utilisez les informations de la bibliothèque de types de Transformer pour créer des scripts de macro dans VB.NET, C# ou pour créer des fichiers d'en-tête et des fichiers d'implémentation dans Visual C++. Vous pouvez aussi employer d'autres applications, telles que Word ou Excel, pour visualiser la bibliothèque de types par l'intermédiaire de l'Editeur Visual Basic fourni avec ces produits.

Les bibliothèques de types et l'automatisation OLE ne fonctionnent que dans les environnements Windows. Elles ne fonctionnent pas dans d'autres environnements tels qu'UNIX.

Les procédures ci-après utilisent Microsoft Visual Studio en tant qu'environnement de développement intégré (IDE).

### Procédure

1. Ouvrez un projet.
2. Dans le menu **Projet**, cliquez sur **Ajouter des références**.
3. Cliquez sur **Ajouter** puis, dans la boîte de dialogue **Ajouter une référence**, sélectionnez **IBM Cognos Transformer Application Control**.

**Remarque** : Si cette entrée ne figure pas dans la liste, cliquez sur le bouton **Parcourir**, puis sélectionnez le fichier TransformerSDK.dll, situé dans le dossier bin de l'emplacement d'installation de Transformer. La bibliothèque TransformerSDK.dll est fournie avec l'installation de Transformer.

4. Dans le menu **Afficher**, cliquez sur **Explorateur d'objets**.
5. Dans la liste Bibliothèque (en haut à gauche de la fenêtre), cliquez sur **Interop.TransformerSDKLib**.
6. Recherchez l'objet ou la collection de votre choix, puis sélectionnez les méthodes et les propriétés correspondantes.
7. Utilisez ces méthodes et ces propriétés dans votre macro.

---

## Présentation de la bibliothèque de types pour Visual C++ et Visual Studio 2009 C#

Une bibliothèque de types est un fichier binaire qui contient des interfaces de classe et des listes de valeurs (énumérateurs) reconnus par un serveur d'automatisation OLE. Les bibliothèques de types sont prises en charge par les outils de développement, tels que Microsoft Visual Basic, Visual C# et Visual C++. Les bibliothèques de types qui sont intégrées à l'environnement de développement vous permettent d'extraire des informations d'automatisation pendant la compilation et l'exécution d'une application.

La bibliothèque de types de Transformer contient des informations relatives aux objets et aux collections, ainsi qu'à leurs propriétés et méthodes. Elle offre aussi des fonctionnalités telles que :

- la vérification des erreurs à la compilation plutôt qu'à l'exécution,
- la liaison précoce des méthodes et des propriétés plutôt que la liaison tardive, ce qui augmente la vitesse d'exécution,
- des fonctions d'aide avancées lors de la création de scripts.

Utilisez les informations de la bibliothèque de types de Transformer pour créer des scripts de macro dans VB.NET, C# ou pour créer des fichiers d'en-tête et des fichiers d'implémentation dans Visual C++. Vous pouvez aussi employer d'autres applications, telles que Word ou Excel, pour visualiser la bibliothèque de types par l'intermédiaire de l'Editeur Visual Basic fourni avec ces produits.

Les bibliothèques de types et l'automatisation OLE ne fonctionnent que dans les environnements Windows. Elles ne fonctionnent pas dans d'autres environnements tels qu'UNIX.

Les procédures ci-après utilisent Microsoft Visual Studio en tant qu'environnement de développement intégré (IDE).

### Procédure

1. Ouvrez Visual Studio.
2. Ouvrez un projet.

Ce projet doit permettre l'ajout de bibliothèques de types.

3. Dans la fenêtre **Explorateur de solutions**, cliquez sur **Références**, cliquez avec le bouton droit de la souris et sélectionnez **Ajouter une référence**.

**Astuce :** Vous pouvez également cliquer sur **Ajouter une référence** dans le menu **Projet**.

4. Dans la boîte de dialogue **Ajouter une référence**, cliquez sur **Parcourir** et sélectionnez le fichier TransformerSDK.dll, situé dans le dossier bin dans lequel vous avez installé le kit Transformer Software Development Kit. La bibliothèque TransformerSDK.dll est fournie avec l'installation de Transformer.

Pour consulter la référence que vous avez ajoutée, dans le menu **Outils**, cliquez sur **Explorateur d'objets OLE/COM**. Sous **Contrôles**, recherchez **IBM Cognos Transformer Application Control**.

Pour afficher des détails supplémentaires, cliquez avec le bouton droit de la souris et sélectionnez l'option **Afficher les informations de type** dans le menu de raccourci.

---

## Présentation de la bibliothèque de types de Transformer

Une bibliothèque de types est un fichier binaire qui contient des interfaces de classe et des listes de valeurs (énumérateurs) reconnus par un serveur d'automatisation OLE. Les bibliothèques de types sont prises en charge par les outils de développement, tels que Microsoft Visual Basic, Visual C# et Visual C++. Les bibliothèques de types qui sont intégrées à l'environnement de développement vous permettent d'extraire des informations d'automatisation pendant la compilation et l'exécution d'une application.

La bibliothèque de types de Transformer contient des informations relatives aux objets et aux collections, ainsi qu'à leurs propriétés et méthodes. Elle offre aussi des fonctionnalités telles que :

- la vérification des erreurs à la compilation plutôt qu'à l'exécution,
- la liaison précoce des méthodes et des propriétés plutôt que la liaison tardive, ce qui augmente la vitesse d'exécution,
- des fonctions d'aide avancées lors de la création de scripts.

Utilisez les informations de la bibliothèque de types de Transformer pour créer des scripts de macro dans VB.NET, C# ou pour créer des fichiers d'en-tête et des fichiers d'implémentation dans Visual C++. Vous pouvez aussi employer d'autres applications, telles que Word ou Excel, pour visualiser la bibliothèque de types par l'intermédiaire de l'Editeur Visual Basic fourni avec ces produits.

Les bibliothèques de types et l'automatisation OLE ne fonctionnent que dans les environnements Windows. Elles ne fonctionnent pas dans d'autres environnements tels qu'UNIX.

Les procédures ci-après utilisent Microsoft Visual Studio en tant qu'environnement de développement intégré (IDE).

### Procédure

1. Ouvrez Visual Studio.
2. Ouvrez un projet.  
Ce projet doit permettre l'ajout de bibliothèques de types.
3. Dans le menu **Projet**, cliquez sur **Ajouter une classe**, puis sélectionnez **Catégories MFC/Classe MFC à partir du contrôle ActiveX** dans la liste **Modèles** de la boîte de dialogue **Ajouter une classe**.  
Dans **Assistant Ajout d'une classe à partir d'un contrôle ActiveX**, cliquez sur **Catégories MFC/Classe MFC à partir du contrôle ActiveX**.
4. Dans la zone **Contrôles ActiveX disponibles**, sélectionnez **IBM Cognos Transformer Application Control<version#>**, où **version#** correspond au numéro de version du kit SDK (Software Development Kit) Transformer.  
Si l'option **A partir du registre** est sélectionnée, le fichier TransformerSDK.dll doit être spécifié.
5. Sélectionnez les interfaces souhaitées et cliquez sur **Terminer**.
6. Pour afficher les classes sélectionnées et leurs objets dans l'espace de travail, cliquez sur **Affichage de classes**.
7. Pour afficher les définitions d'objets dans les différentes classes de wrapper, cliquez sur **Affichage de fichiers**.

---

## Présentation des méthodes

Dans l'automatisation OLE, les méthodes permettent de créer, de modifier et de supprimer des objets.

Certaines méthodes fournies avec Transformer se comportent comme des sous-programmes et d'autres comme des fonctions. Vous pouvez déterminer le type de méthode de l'une des façons suivantes :

- Si vous utilisez un outil de développement Visual Studio, recherchez la méthode dans l'Explorateur d'objets.
- Utilisez l'outil Oleview. Pour accéder à Oleview, dans le menu **Démarrer**, cliquez sur **Exécuter** et saisissez **oleview**. Cliquez sur **OK**. Sous **Contrôles**, recherchez **IBM Cognos Transformer Application Control**. Pour afficher les informations de type, cliquez avec le bouton droit de la souris et sélectionnez l'option **Afficher les informations de type** dans le menu de raccourci.

Les informations de type TransformerSDK affichent les paramètres entre parenthèses, à la fois pour les méthodes de sous-programme et les méthodes de fonction, mais celles-ci ne sont pas toujours nécessaires. Les méthodes qui définissent des sous-programmes ne nécessitent pas de parenthèses et ne renvoient pas de valeur. En revanche, les fonctions requièrent des parenthèses et retournent des valeurs. Cela varie également en fonction du langage d'implémentation utilisé et de la présence ou non d'objets génériques.

---

## Présentation des listes de valeurs et des constantes

La plupart des propriétés et des méthodes de Transformer OLE utilisent un ensemble de types de données énumérées ou des listes de valeurs. Chaque liste de valeurs contient un ensemble de constantes qui définissent l'action de la propriété ou de la méthode applicable.

Les noms des listes de valeurs sont précédés du préfixe "xtr". Les constantes proprement dites sont précédées du préfixe "tr". Par exemple, les valeurs xtrCubeCreation utilisées pour définir les options de création des cubes sont :

- trCubeCreationDefault
- trCubeCreationON
- trCubeCreationOFF

Vous pouvez utiliser des constantes des listes de valeurs dans les opérations de définition et de retour. Cet exemple renvoie une constante de la liste des valeurs xtrObjectType :

```
If objDimension1.ObjectType = xtrObjectType.trDateDimension
Then
.
.
End If
```

Cet exemple utilise une constante de la liste des valeurs xtrObjectType pour ajouter un objet DateDimension à une collection :

```
objDimCollection.Add(xtrObjectType.trDateDimension)
```

Notez que si vous avez précédemment utilisé l'Editeur IBM CognosScript pour créer des macros Transformer OLE, le fichier TranConst.inc a été mis à jour de

façon à être compatible avec Cognos TransformerSDK. En revanche, cette application d'éditeur n'est plus utilisée. Ce fichier ne peut pas être utilisé avec VB.NET, C# ou C++.

Une copie du fichier TranConst.inc est disponible à l'emplacement d'installation de Transformer, dans le dossier *installation\_location*\ templates\cogtr\TransformerSDK.

Voici un exemple d'instruction Include faisant référence au fichier TranConst.inc.

```
$Include "TranConst.inc"
```

Bien que toutes les constantes possèdent des équivalents numériques, nous vous recommandons de les référencer à l'aide de leur nom afin d'éviter tout problème pendant la mise à niveau de TransformerSDK. Lorsque vous faites appel à une technologie .NET, la notation des éléments d'énumération "type.enumeration" doit être utilisée.

---

## Grille hiérarchique des objets de Transformer

Lorsque vous concevez un modèle, vous devez créer vos collections et vos objets, puis leur affecter des valeurs dans un ordre hiérarchique.

La grille ci-après répertorie toutes les collections et tous les objets, et affiche leur ordre dans la hiérarchie.



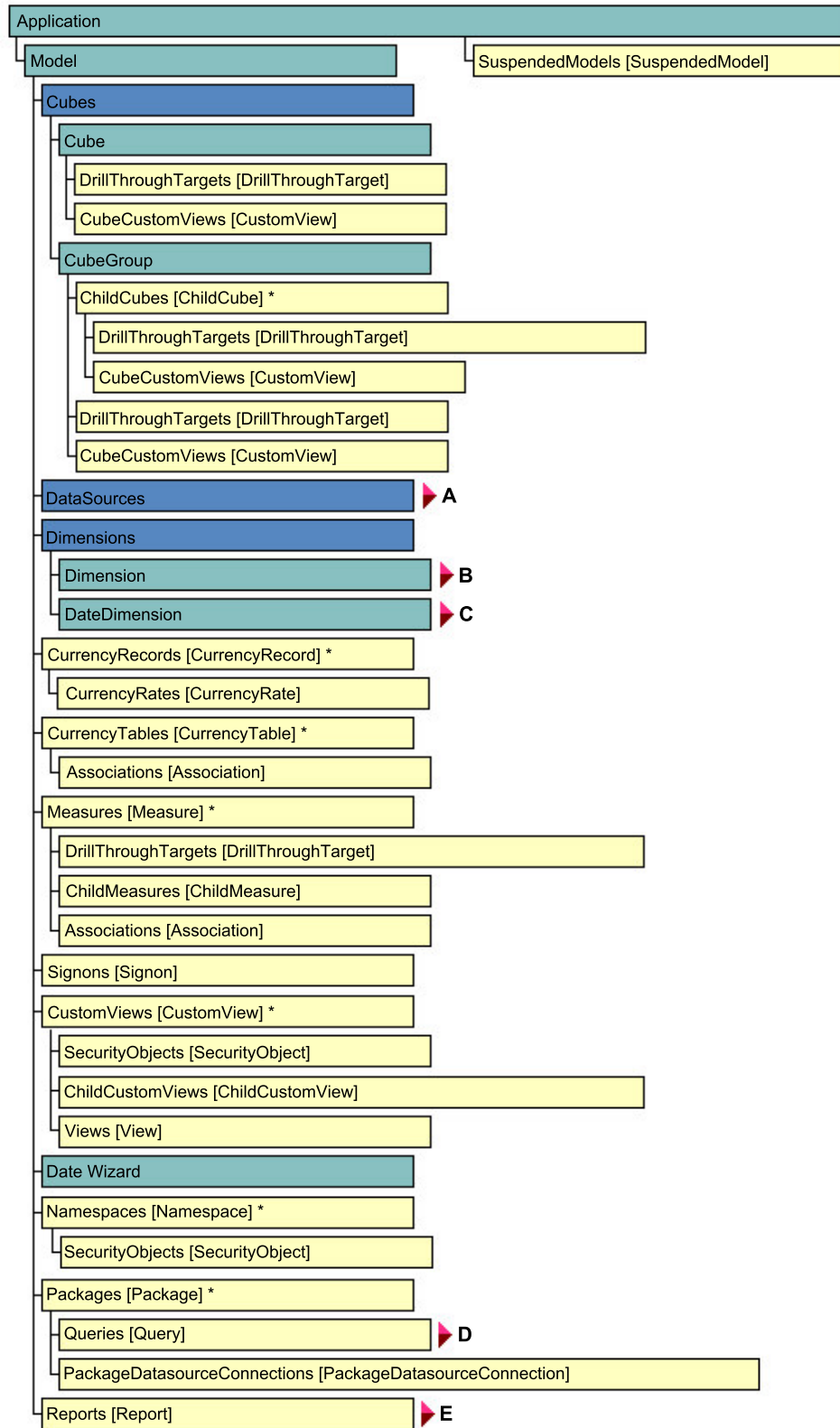


Figure 1. Tous les objets

- ■ Objets
- ■ Collections

- Collections et objets

L'astérisque (\*) s'applique à l'objet et non à la collection.

---

## Grille hiérarchique A des objets de Transformer

La grille suivante présente la hiérarchie DataSources.

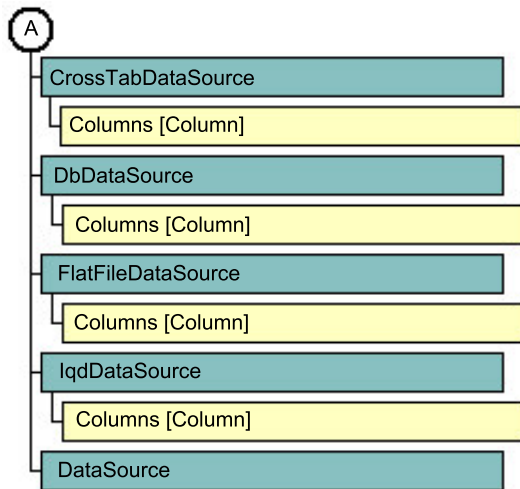


Figure 2. Hiérarchie DataSources

- Objets
- Collections
- Collections et objets

## Grille hiérarchique B des objets de Transformer

La grille suivante présente la hiérarchie Dimension.

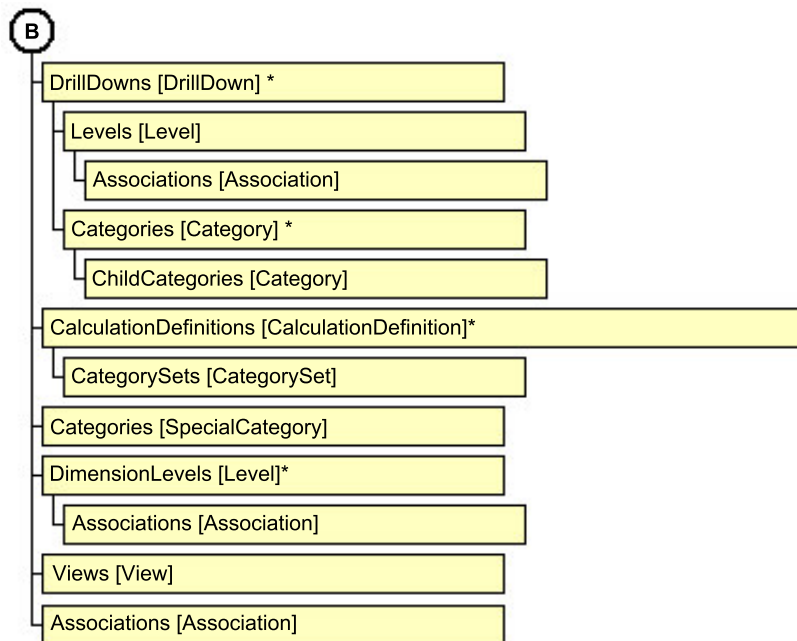


Figure 3. Hiérarchie Dimension

- ■ Objets
- ■ Collections
- ■ Collections et objets

L'astérisque (\*) s'applique à l'objet et non à la collection.

---

## Grille hiérarchique C des objets de Transformer

La grille suivante présente la hiérarchie DateDimension.

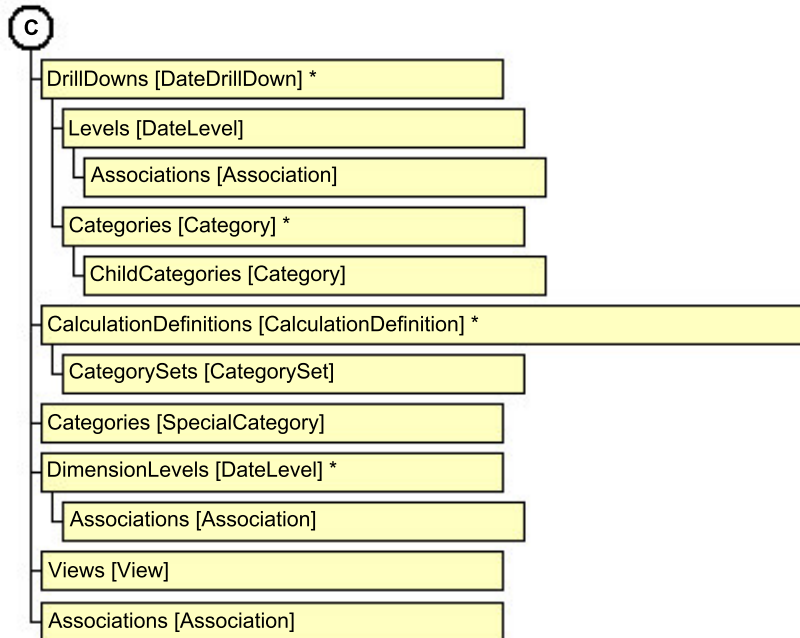


Figure 4. Hiérarchie DateDimension

- Objets
- Collections
- Collections et objets

L'astérisque (\*) s'applique à l'objet et non à la collection.

---

## Grille hiérarchique D des objets de Transformer

La grille suivante présente la hiérarchie Query.

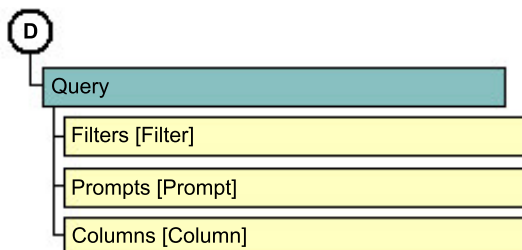


Figure 5. Hiérarchie Query

- Objets
- Collections
- Collections et objets

---

## Grille hiérarchique E des objets de Transformer

La grille suivante présente la hiérarchie Report.

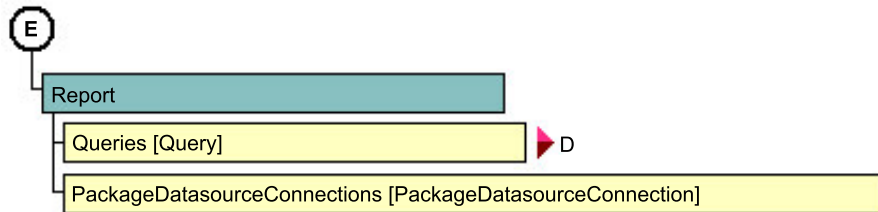





Figure 6. Hiérarchie Report

-  Objets
-  Collections
-  Collections et objets

---

## Paramètres préférentiels du kit de développement de logiciels

Veillez à définir les préférences du kit SDK (Software Development Kit) de façon à garantir un bon fonctionnement de toutes les applications. Ces préférences peuvent être définies à l'aide des propriétés de l'objet Application.

---

## Modèles IBM Cognos Series 7

Notez qu'avant d'utiliser les modèles IBM Cognos Series 7 avec cette version de l'automatisation OLE, vous devez les mettre à niveau.

Pour plus d'informations sur la mise à niveau des modèles IBM Cognos Series 7, voir le *Guide d'utilisation* de Transformer.



---

## Chapitre 3. Collections

Les tableaux ci-dessous répertorient toutes les collections d'automatisation OLE de Transformer.

Objet	Description
Collection Associations	Contient tous les objets Association d'un objet donné.
Collection CalculationDefinitions	Contient tous les objets CalculationDefinition d'une dimension donnée.
Collection Categories	Regroupe des objets Category ou SpecialCategory.
Collection CategorySets	Contient tous les objets CategorySet qui sont employés par un objet CalculationDefinition pour calculer des valeurs.
Collection ChildCubes	Regroupe des objets ChildCube.
Collection Columns	Regroupe des objets Column dans une source de données.
Collection CubeCustomViews	Contient des objets CustomView qui ont accès à un objet Cube, CubeGroup ou ChildCube spécifique.
Collection Cubes	Regroupe tous les objets Cube et CubeGroup dans un modèle.
Collection CurrencyRates	Regroupe des objets CurrencyRate.
Collection CurrencyRecords	Contient tous les objets CurrencyRecord d'un modèle.
Collection CurrencyTables	Contient tous les objets CurrencyTable d'un modèle.
collection CustomViews	Représente une collection d'objets CustomView dans un modèle.
Collection DataSources	Contient toutes les sources de données d'un modèle.
Collection DimensionLevels	Contient une liste en lecture seule répertoriant les niveaux uniques de la dimension associée.
Collection Dimensions	Contient tous les objets Dimension et DateDimension d'un modèle.
Collection DrillDowns	Contient des objets DrillDown ou DateDrillDown, mais pas les deux.
Collection DrillThroughTargets	Groupes relatifs aux objets DrillThroughTarget.
Collection Filters	Contient tous les objets Filter d'une requête.
collection LevelCategories	Contient une collection de catégories pour un niveau spécifique.
Collection LevelDrillDowns	Représente une collection en lecture seule d'objets représentant des branches de niveaux.

Objet	Description
Collection Levels	Regroupe des objets de niveau ou DateLevel, mais pas les deux.
Collection Measures	Contient tous les objets Measure d'un modèle.
Collection Names	Contient un groupe en lecture seule répertoriant des objets Name.
Collection Namespaces	Contient tous les objets Namespace d'un modèle.
Collection PackageDatasourceConnections	Contient tous les objets PackageDatasourceConnection.
Collection Packages	Contient tous les objets Package d'un modèle.
Collection Prompts	Contient tous les objets Prompt d'une requête.
Collection Queries	Contient tous les objets Query d'un pack ou d'un rapport.
Collection Reports	Regroupe des objets Report associés.
Collection SecurityObjects	Contient tous les objets SecurityObject d'un modèle.
Collection Signons	Regroupe des objets de code d'accès associés.
Collection SuspendedModels	Contient tous les objets SuspendedModel disponibles pour l'application.
Collection Views	Regroupe des objets View.

---

## Collection Associations

La collection Associations contient tous les objets Association d'un objet donné.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Association.

La plupart des objets d'un modèle, tels que les niveaux et les dimensions, conservent des associations avec des données externes par l'intermédiaire d'un ou plusieurs objets Association. L'association fournit des valeurs pour les données sources, ainsi que pour les propriétés qui représentent des libellés, des descriptions, des noms de tri, etc.

Pour retourner cette collection, utilisez la propriété Associations.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Objects)	Ajoute un objet Association à la collection.
Méthode Item	Sélectionne un objet Association spécifique dans la collection.



Méthode	Description
Méthode Remove	Supprime un objet Association spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Association de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()
```

---

## Collection CalculationDefinitions

La collection CalculationDefinitions contient tous les objets CalculationDefinition d'une dimension donnée.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CalculationDefinition.

Les objets CalculationDefinition sont constitués d'expressions complexes qui incluent dans le calcul les valeurs d'une ou plusieurs catégories.

Pour retourner cette collection, utilisez la propriété CalculationDefinitions.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Objects)	Ajoute un objet CalculationDefinition à la collection.
Méthode Item	Sélectionne un objet CalculationDefinition spécifique dans la collection.
Méthode Remove	Supprime un objet CalculationDefinition spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CalculationDefinition de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
calcDef = dimension.CalculationDefinitions.Add()
```

---

## Collection Categories

La collection Categories regroupe des objets Category ou des objets SpecialCategory.

### Explication

Utilisez la propriété Categories d'un objet DrillDown pour retourner une collection de catégories ordinaires.

Utilisez la propriété ChildCategories d'un objet Category pour retourner une collection de catégories descendantes.

Utilisez la propriété Categories d'un objet Dimension ou DateDimension pour retourner une collection de catégories spéciales.

L'une ou l'autre collection peut contenir des catégories calculées, qui sont associées à des catégories ordinaires ou spéciales.

Pour générer des catégories pour toutes les dimensions, utilisez la méthode GenerateCategories. Vous pouvez modifier manuellement ces collections. Utilisez, par exemple, la méthode Add pour ajouter des objets SpecialCategory à une collection ou définissez la propriété Inclusion pour exclure des objets Category spécifiques.

Vous pouvez créer manuellement une collection Categories pendant l'élaboration de niveaux manuels. Cela s'avère utile quand le nombre d'objets Category d'une collection générée est trop élevé pour pouvoir être affiché correctement dans PowerPlay.

Un objet CalculationDefinition utilise la collection Categories d'un objet CategorySet pendant la création d'une expression. Utilisez la méthode Add de la collection Categories chaque fois que vous ajoutez une catégorie à l'objet CategorySet.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Categories)	Ajoute un objet Category ou SpecialCategory à la collection.
Méthode Item	Sélectionne un objet Category ou SpecialCategory spécifique dans la collection.
Méthode Remove	Supprime un objet Category ou SpecialCategory spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.

Propriété	Description
Propriété Count	Retourne le nombre d'objets Category ou SpecialCategory de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
objSpecCategory = _
objModel.Dimensions("Time").Categories.Add(xtrObjectType.trSpecialCategory)
```

---

## Collection CategorySets

La collection CategorySets contient tous les objets CategorySet qui sont employés par un objet CalculationDefinition pour calculer des valeurs.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CategorySet. Chaque objet CategorySet contient, à son tour, une collection Categories.

Pour retourner cette collection, utilisez la propriété CategorySets.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Objects)	Ajoute un objet CategorySet à la collection.
Méthode Item	Sélectionne un objet CategorySet spécifique dans la collection.
Méthode Remove	Supprime un objet CategorySet spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CategorySet de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
catSet = calcDef.CategorySets.Add()
```

---

## Collection ChildCubes

La collection ChildCubes regroupe des objets ChildCube.

### Explication

Cette collection permet de sélectionner un objet ChildCube. Chaque cube de cette collection fait partie d'un objet CubeGroup et représente une catégorie unique dans le niveau associé à CubeGroup.

Pour retourner cette collection, utilisez la propriété ChildCubes.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet ChildCube spécifique dans la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets ChildCube de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)
objChildCube = objCubesByRegion.ChildCubes("Central Europe")
```

---

## Collection Columns

La collection Columns regroupe des objets Column dans une source de données.

## Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Column. Les données des sources de données de Transformer sont organisées dans des colonnes qui agissent comme source des valeurs des niveaux.

Pour retourner cette collection, utilisez la propriété Columns d'un objet source de données.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Objects)	Ajoute un objet Column à la collection.
Méthode Item	Sélectionne un objet Column spécifique dans la collection.
Méthode Remove	Supprime un objet Column spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Column de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
objColumn = objDataSource.Columns(1)
```

---

## Collection CubeCustomViews

La collection CubeCustomViews contient des objets CustomView qui sont associés à un objet Cube, CubeGroup ou ChildCube spécifique.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CustomView. Chaque objet Cube ou ChildCube définit une collection CubeCustomView. Pour retourner cette collection, utilisez la propriété CubeCustomViews.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (collection CubeCustomViews)	Ajoute un objet CustomView à la collection.
Méthode Item	Sélectionne un objet CustomView spécifique dans la collection.
Méthode Remove	Supprime un objet CustomView spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CustomView de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
cube = model.Cubes.Item("Sales and Marketing")
cube.CubeCustomViews.Add(custom_view)
```

---

## Collection Cubes

La collection Cubes regroupe tous les objets Cube et CubeGroup dans un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Cube ou CubeGroup. L'objet Cube contient les collections Reports et CubeCustomViews associées. L'objet CubeGroup contient les collections ChildCubes, DrillThroughTargets et CubeCustomViews associées.

Pour retourner cette collection, utilisez la propriété Cubes.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add (Objects)	Ajoute un objet Cube ou CubeGroup à la collection.
Méthode Item	Sélectionne un objet Cube ou CubeGroup spécifique dans la collection.

Méthode	Description
Méthode Remove	Supprime un objet Cube ou CubeGroup spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Cube et CubeGroup de la collection.
Propriété Parent	Retourne l'objet parent.

---

## Collection CurrencyRates

La collection CurrencyRates regroupe des objets CurrencyRate.

### Explication

Cette collection permet de sélectionner un objet CurrencyRate. Chaque objet représente le taux de conversion utilisé pour convertir les devises.

La conversion des devises exige l'emploi d'une collection CurrencyRates, CurrencyRecords et CurrencyTables. Pour retourner cette collection, utilisez la propriété CurrencyRates.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet CurrencyRate spécifique dans la collection.
Méthode Remove	Supprime un objet CurrencyRate spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CurrencyRate de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()
objCurrencyRate = objCurrencyRecord.CurrencyRates(intX)
```

---

## Collection CurrencyRecords

La collection CurrencyRecords contient tous les objets CurrencyRecord d'un modèle.

## Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CurrencyRecord. Chaque objet de la collection représente une ligne d'informations sur la devise qui est utilisée pour formater et afficher les devises dans PowerPlay.

La conversion des devises exige l'emploi d'une collection CurrencyRates, CurrencyRecords et CurrencyTables.

Pour retourner cette collection, utilisez la propriété CurrencyRecords.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet CurrencyRecord à la collection.
Méthode Item	Sélectionne un objet CurrencyRecord spécifique dans la collection.
Méthode Remove	Supprime un objet CurrencyRecord spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CurrencyRecord de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()
```

---

## Collection CurrencyTables

La collection CurrencyTables contient tous les objets CurrencyTable d'un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CurrencyTable.

Vous pouvez utiliser des sources de données externes pour remplir une table de devises, ou créer et insérer des enregistrements de devise et des taux de conversion. Ces deux méthodes nécessitent des données qui définissent

- la date de conversion,
- Code ISO pays ou région
- le taux de conversion,
- le libellé de la devise (facultatif).

Transformer utilise les informations de la table des devises pour effectuer les conversions appropriées quand un utilisateur affiche des cubes dans PowerPlay. La conversion monétaire requiert au moins la présence d'une dimension de temps dans le modèle. Transformer stocke les informations relatives aux devises au même niveau de détail que celui de la dimension de temps (par défaut, au niveau du mois). Vous pouvez éventuellement l'ajuster à un niveau plus élevé.

Les conversions de devises de l'UEM exigent deux tables. Dans une table, la propriété CurrencyTableType a pour valeur trCurrencyTableBase et dans l'autre, trCurrencyTableEuro. Une collection CurrencyTables ne peut contenir qu'un seul type de table.

La conversion des devises exige l'emploi d'une collection CurrencyRates, CurrencyRecords et CurrencyTables. Pour retourner cette collection, utilisez la propriété CurrencyTables.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet CurrencyTable à la collection.
Méthode Item	Sélectionne un objet CurrencyTable spécifique dans la collection.
Méthode Remove	Supprime un objet CurrencyTable spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CurrencyTable de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objBaseTable = objModel.CurrencyTables.Add()
```

---

## collection CustomViews

La collection CustomViews représente une collection d'objets CustomView dans un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CustomView. L'objet CustomView contient les collections SecurityObjects et Views associées.



Pour retourner cette collection, utilisez la propriété CustomViews ou ChildCustomViews.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet CustomView à la collection.
Méthode Item	Sélectionne un objet CustomView spécifique dans la collection.
Méthode Remove	Supprime un objet CustomView spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets CustomView de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
custom_view = model.CustomViews.Add()  
custom_view = model.CustomViews.Add()
```

---

## Collection DataSources

La collection DataSources contient toutes les sources de données d'un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer une source de données. Cette collection peut inclure les objets suivants :

- CrossTabDataSource
- DataSource
- DbDataSource,
- FlatFileDataSource
- IqdDataSource

Pour retourner cette collection, utilisez la propriété DataSources.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet à une collection qui contient des objets de plusieurs types.
Méthode Item	Sélectionne une source de données spécifique dans la collection.
Méthode Remove	Supprime une source de données spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.

Propriété	Description
Propriété Count	Retourne le nombre d'objets DataSource de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
```

---

## Collection DimensionLevels

La collection DimensionLevels contient une liste en lecture seule répertoriant les niveaux uniques de la dimension associée.

### Explication

Cette collection permet de sélectionner un objet de niveau à partir d'une dimension.

Lorsqu'une dimension possède des branches de niveaux secondaires, un niveau individuel peut être inclus à plusieurs reprises dans la dimension. Cette collection ne contient toutefois qu'une seule référence à ce niveau, même s'il apparaît à plusieurs reprises dans la dimension.

Pour retourner cette collection, utilisez la propriété DimensionLevels.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet spécifique dans la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objLevel = objModel.Dimensions("Products").DimensionLevels(1)
```

---

## Collection Dimensions

La collection Dimensions contient tous les objets Dimension et DateDimension d'un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Dimension ou DateDimension.

Quand vous créez des dimensions à l'aide de la méthode DoAutoDesign, Transformer ajoute automatiquement des dimensions à cette collection sur base des relations existantes dans les sources de données. L'objet DateWizard ajoute des objets DateDimension à cette collection.

Pour retourner cette collection, utilisez la propriété Dimensions.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Dimension ou DateDimension à la collection.
Méthode Item	Sélectionne un objet Dimension ou DateDimension spécifique dans la collection.
Méthode Remove	Supprime un objet Dimension ou DateDimension spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Dimension et DateDimension de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objDimensions = objModel.Dimensions
```

---

## Collection DrillDowns

La collection DrillDowns contient des objets DrillDown ou DateDrillDown, mais pas les deux.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet DrillDown ou DateDrillDown.

Transformer ajoute automatiquement un objet DrillDown ou DateDrillDown à chaque collection DrillDowns lors de la création de la dimension parent. Cet objet représente la branche principale de niveaux et ne peut pas être supprimé. Tout autre objet DrillDown ou DateDrillDown de la collection représente une branche de niveaux secondaire.

Utilisez la méthode CreateAlternateDrillDown de l'objet de niveau pour ajouter des objets DrillDown ou DateDrillDown à la collection.

Pour retourner cette collection, utilisez la propriété DrillDowns.

Méthode	Description
Méthode Item	Sélectionne un objet DrillDown ou DateDrillDown spécifique dans la collection.

Méthode	Description
Méthode Remove	Supprime un objet DrillDown ou DateDrillDown spécifique de la collection. Ne s'applique pas au premier objet de la collection. En cas d'application au premier objet, une exception est générée.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets DrillDown ou DateDrillDown de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objDrill = objModel.Dimensions("Retailers").DrillDowns(2)
```

---

## Collection DrillThroughTargets

La collection DrillThroughTargets regroupe les objets DrillThroughTarget connexes.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet DrillThroughTarget.

La documentation des éditions précédentes indique qu'un objet DrillThroughTarget représente un objet d'accès au détail, tel qu'un rapport Impromptu (.imr), un fichier de définition de requête d'Impromptu (.iqd), un rapport PowerPlay (.ppr), un fichier de PowerCube (.mdc) ou un fichier de script de macro (.mac). Pour certaines opérations, telles que l'extraction de nombres ou d'autres informations relatives aux objets cible d'accès au détail, vous devez cependant recourir aux scripts MDL.

Pour retourner la collection DrillThroughTargets pour un objet Measure, Cube ou CubeGroup, utilisez la propriété DrillThroughTargets. Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Measure, vous limitez la fonctionnalité d'accès au détail à cette seule mesure. Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Cube ou CubeGroup, la fonctionnalité d'accès au détail est disponible à n'importe quel point du cube ou des cubes enfants.

Dans le cas de modèles qui utilisent des fichiers de définition de requête d'Impromptu (.iqd) comme source de données, Transformer ajoute automatiquement le rapport Impromptu (.imr) correspondant à la collection DrillThroughTargets pour chaque mesure.

Vous devez affecter la valeur True à la propriété AllowDrillThrough avant de pouvoir accéder à la fonction d'accès au détail.

Méthode	Description
Méthode Add (DrillThroughTargets)	Ajoute un objet DrillThroughTarget à la collection.
Méthode Item	Sélectionne un objet DrillThroughTarget spécifique dans la collection.
Méthode Remove	Supprime un objet DrillThroughTarget spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Report de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
objReport = objMeasure.DrillThroughTargets.Add(strReportPath,
"Default Report")
```

---

## Collection Filters

La collection Filters contient tous les objets Filter d'une requête.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet Filter.

Cette collection est un objet IBM Cognos. Chaque objet Query ou Report d'un modèle peut comporter une collection Filters. Un filtre représente une condition qui doit être remplie afin que les données soient extraites de la source de données. Pour retourner cette collection, utilisez la propriété Filters.

Utilisez la méthode Add pour ajouter un nouvel objet Filter à la collection. Pour enlever un objet Filter, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Filter à la collection.
Méthode Item	Sélectionne un objet Filter spécifique dans la collection.
Méthode Remove	Supprime un objet Filter spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Filter de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
new_package = model.Packages.Add()
new_query = new_package.Queries.Add()
new_filter = new_query.Filters.Add()
```

---

## collection LevelCategories

La collection LevelCategories représente la collection de catégories d'un niveau spécifique.

### Explication

Utilisez cette collection pour sélectionner les catégories associées à chaque niveau.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet Category spécifique dans la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Category présents dans la collection.
Propriété Parent	Retourne l'objet parent.

---

## Collection LevelDrillDowns

La collection LevelDrillDowns représente une collection en lecture seule d'objets représentant des branches de niveaux.

### Explication

Cette collection permet de sélectionner un objet DrillDown ou DateDrillDown dans une dimension.

Comme la plupart des niveaux possèdent une seule branche de niveaux, cette collection ne renferme généralement qu'un seul objet DrillDown ou DateDrillDown. Quand un niveau correspond au niveau de convergence de deux branches de niveaux ou davantage, la collection possède un nombre d'entrées équivalent.

Une collection LevelDrillDowns peut contenir des objets DrillDown ou DateDrillDown, mais pas les deux.

Pour retourner cette collection, utilisez la propriété LevelDrillDowns.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet DrillDown ou DateDrillDown spécifique dans la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets DrillDown ou DateDrillDown de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
objLevel = objDateDrillDown.Levels("Year")  
Select Case objLevel.LevelDrillDowns(1).Categories(intX).KeyName
```

---

## Collection Levels

La collection Levels regroupe des objets de niveau ou DateLevel, mais pas les deux.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet de niveau ou DateLevel.

Chaque objet Dimension et DateDimension d'un modèle contient une collection Levels qui représente les étapes de la branche de niveaux utilisées pour retrouver des informations dans PowerPlay. Transformer crée automatiquement des collections Levels quand vous utilisez la méthode DoAutoDesign.

Pour retourner cette collection, utilisez la propriété Levels.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet de niveau ou DateLevel à la collection.
Méthode Item	Sélectionne un objet de niveau ou DateLevel spécifique dans la collection.
Méthode Remove	Supprime un objet de niveau ou DateLevel spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets de niveau ou DateLevel de la collection.

## Exemples

```
objTimeDimension = objModel.Dimensions.Item("Time")objLevel
=
objTimeDimension.DrillDowns.Item(1).Levels.Item("Month")
```

---

## Collection Measures

La collection Measures contient tous les objets Measure d'un modèle.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Measure.

Chaque objet Measure de la collection représente des valeurs numériques qui agissent comme des indicateurs de performances dans PowerPlay. Les mesures sont toujours quantifiables. Les mesures Revenus, Revenus par employé et Marge de profit sont valides. Transformer crée automatiquement la collection quand vous utilisez la méthode DoAutoDesign.

Une collection Measures peut contenir des mesures ordinaires, des mesures calculées ou des mesures de nombre de catégories. Pour retourner cette collection, utilisez la propriété Measures.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Measure à la collection.
Méthode Item	Sélectionne un objet Measure spécifique dans la collection.
Méthode Remove	Supprime un objet Measure spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Measure de la collection.
Propriété Parent	Retourne l'objet parent.



## Exemples

`objMeasures = objModel.Measures`

---

## Collection Names

La collection Names contient un groupe en lecture seule répertoriant des objets Name.

### Explication

Cette collection permet de référencer un groupe de chaînes représentant des noms ou des messages. Une collection Names est renvoyée par la méthode CheckModel (objet Model).

Chaque objet Name de cette collection contient une chaîne de message complète. Vous pouvez utiliser la valeur de retour de la propriété Name pour lire les résultats de la procédure CheckModel.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet Name dans la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Name de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

`MsgBox(objModel.CheckModel(intX).Name)`

---

## Collection Namespaces

La collection Namespaces contient tous les objets Namespace d'un modèle.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet Namespace.

L'objet IBM Cognos représente une instance d'un fournisseur d'authentification qui autorise le contrôle d'accès et l'authentification. Pour retourner cette collection, utilisez la propriété Namespaces. Le nombre de la collection est égal à zéro si le modèle n'utilise pas Namespaces. Pour utiliser cet objet, vous devez utiliser un serveur IBM Cognos. Cela donne accès aux informations sur les utilisateurs et groupes membres de l'objet SecurityObjects.

Utilisez la méthode Add pour ajouter un nouvel objet Namespace à la collection. Pour enlever un objet Namespace, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Namespace à la collection.
Méthode Item	Sélectionne un objet Namespace spécifique dans la collection.
Méthode Remove	Supprime un objet Namespace spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Namespace de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
new_namespace = model.Namespaces.Add()
```

---

## Collection PackageDatasourceConnections

La collection PackageDatasourceConnections contient tous les objets PackageDatasourceConnection.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet PackageDatasourceConnection.

L'objet IBM Cognos requiert l'installation et le fonctionnement du serveur IBM Cognos. Les objets PackageDatasourceConnection sont créés via IBM Cognos Connection. Le nombre de la collection est égal à zéro si le modèle n'utilise pas Packages.

Pour retourner cette collection, utilisez la propriété PackageDatasourceConnections.

Utilisez la méthode Add pour ajouter un nouvel objet PackageDatasourceConnection à la collection. Pour enlever un objet PackageDatasourceConnection, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet PackageDatasourceConnection à la collection.

Méthode	Description
Méthode Item	Sélectionne un objet PackageDatasourceConnection spécifique dans la collection.
Méthode Remove	Supprime un objet PackageDatasourceConnection spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets PackageDatasourceConnection de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
package = model.Packages.Add()
connection = package.PackageDatasourceConnections.Add()
```

---

## Collection Packages

La collection Packages contient tous les objets Package d'un modèle.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet Package.

IBM Cognos requiert l'installation et le fonctionnement du serveur IBM Cognos. Par définition, un pack peut être un sous-ensemble d'un modèle Transformer mis à disposition du serveur IBM Cognos. Le nombre de la collection est égal à zéro si le modèle n'utilise pas Packages.

Pour retourner cette collection, utilisez la propriété Packages.

Utilisez la méthode Add pour ajouter un nouvel objet Package à la collection. Pour enlever un objet Package, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Package à la collection.
Méthode Item	Sélectionne un objet Package spécifique dans la collection.

Méthode	Description
Méthode Remove	Supprime un objet Package spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Package de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

`packages = model.Packages`

---

## Collection Prompts

La collection Prompts contient tous les objets Prompt d'une requête.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet Prompt.

IBM Cognos requiert l'installation et le fonctionnement du serveur IBM Cognos. Les invites sont associées à des requêtes. Pour retourner cette collection, utilisez la propriété Prompts d'un objet Query.

Utilisez la méthode Add pour ajouter un nouvel objet Prompt à la collection. Pour enlever un objet Prompt, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Prompt à la collection.
Méthode Item	Sélectionne un objet Prompt spécifique dans la collection.
Méthode Remove	Supprime un objet Prompt spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Prompt de la collection.

Propriété	Description
Propriété Parent	Retourne l'objet parent.

## Exemples

```
new_report = model.Reports.Add()
new_query = new_report.Queries.Add()
new_prompt = new_query.Prompts.Add()
```

---

## Collection Queries

La collection Queries contient tous les objets Query d'un pack ou d'un rapport.

### Explication

La collection Queries permet de sélectionner ou de supprimer un objet Query.

IBM Cognos requiert l'installation et le fonctionnement du serveur IBM Cognos. Un objet Query représente une spécification d'un ensemble de données extraites d'une source de données. Les modèles Transformer peuvent comprendre plusieurs requêtes comme sources de données.

Pour retourner cette collection, utilisez la propriété Queries.

Utilisez la méthode Add pour ajouter un nouvel objet Query à la collection. Pour enlever un objet Query, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Méthode ajoutant un objet Query à la collection.
Méthode Item	Sélectionne un objet Query spécifique dans la collection.
Méthode Remove	Supprime un objet Query spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Query de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
new_package = model.Packages.Add()  
new_query = new_package.Queries.Add()
```

---

## Collection Reports

La collection Reports regroupe des objets Report associés.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet Report.

Un rapport est un rapport IBM Cognos qui sert de source de données de type conteneur. Plusieurs requêtes de source de données peuvent être associées à un rapport. Les rapports sont créés par une application IBM Cognos, telle que Query Studio ou Report Studio, à l'aide de packs relationnels ou DMR. Dans la mesure où les rapports sont enregistrés dans IBM Cognos Connection, la présence d'un serveur IBM Cognos est nécessaire pour établir la connexion de source de données. Un rapport peut contenir une ou plusieurs requêtes. Le nombre de la collection est égal à zéro si le modèle n'utilise pas de rapports.

Pour retourner cette collection, utilisez la propriété Reports.

Utilisez la méthode Add pour ajouter un nouvel objet Report à la collection. Pour enlever un objet Report, utilisez la méthode Remove.

Méthode	Description
Méthode Add ()	Ajoute un objet Report à la collection.
Méthode Item	Sélectionne un objet Report spécifique dans la collection.
Méthode Remove	Supprime un objet Report spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Report de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
reports = model.Reports
```

---

## Collection SecurityObjects

La collection SecurityObjects contient tous les objets SecurityObject d'un modèle.

## Explication

Cette collection permet de sélectionner ou de supprimer un objet SecurityObject.

Utilisez la propriété SecurityObjects d'un objet CustomView ou Namespace afin d'extraire cette collection.

Utilisez la méthode Add pour ajouter un nouvel objet SecurityObject à la collection. Pour enlever un objet SecurityObject, utilisez la méthode Remove.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet Report à la collection.
Méthode Item	Sélectionne un objet Report spécifique dans la collection.
Méthode Remove	Supprime un objet Report spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets Report de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
new_namespace = model.Namespaces.Add()  
securityObject = new_namespace.SecurityObjects.Add()
```

---

## Collection Signons

La collection Signons regroupe des objets de code d'accès associés.

### Explication

Cette collection permet d'ajouter, de modifier ou de supprimer un objet de code d'accès.

Chaque objet de la collection régule l'accès aux données dans un objet IqdDataSource. Pour retourner cette collection, utilisez la propriété Signons.

Lorsque vous utilisez un fichier de définition de requête d'Impromptu (.iqd) en tant que source de données, Transformer utilise les informations de ce fichier pour créer automatiquement un nouvel objet de code d'accès à l'intérieur du modèle. Le nouvel objet de code d'accès se voit affecter le nom de la base de données logique présent dans le fichier de définition de requête d'Impromptu (.iqd).

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet de code d'accès à la collection.
Méthode Item	Sélectionne un objet de code d'accès spécifique dans la collection.
Méthode Remove	Supprime un objet de code d'accès spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets de code d'accès de la collection.
Propriété Parent	Retourne l'objet parent.

## Exemples

```
signon = model.Signons.Add()
```

---

## Collection SuspendedModels

La collection SuspendedModels contient tous les objets SuspendedModel disponibles pour l'application.

### Explication

Cette collection permet de sélectionner ou de supprimer un objet SuspendedModel.

Si un modèle se ferme brutalement, notamment lors d'une panne du système, Transformer ajoute à sa collection un objet SuspendedModel pour ce modèle.

Pour retourner cette collection, utilisez la propriété SuspendedModels.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Item	Sélectionne un objet SuspendedModel spécifique dans la collection.
Méthode RemoveSuspendedModel	Supprime un objet SuspendedModel spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets SuspendedModel de la collection.
Propriété Parent	Retourne l'objet parent.



## Exemples

```
If objTransApp.SuspendedModels.Count = 0 Then
```

---

## Collection Views

La collection Views regroupe des objets View.

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet View.

Dans la plupart des organisations, toutes les données du modèle ne doivent pas être mises à la disposition de tous les utilisateurs. Vous pouvez limiter les données notamment en créant une vue, qui est un sous-ensemble des niveaux et des catégories d'une dimension. Vous pouvez aussi regrouper plusieurs vues associées à une dimension dans une collection Views.

Pour retourner cette collection, utilisez la propriété Views.

Ces tables répertorient les méthodes et propriétés connexes.

Méthode	Description
Méthode Add ()	Ajoute un objet View à la collection.
Méthode Item	Sélectionne un objet View spécifique dans la collection.
Méthode Remove	Supprime un objet View spécifique de la collection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Count	Retourne le nombre d'objets View de la collection.
Propriété Parent	Retourne l'objet parent.

### Exemples

```
objDimension = objModel.Dimensions("Retailers")  
objView = objDimension.Views.Add()
```



---

## Chapitre 4. Objets

Le tableau suivant récapitule tous les objets d'automatisation OLE de Transformer :

Objet	Description
Objet Application	Représente une instance de l'application Transformer.
Objet Association	Définit la relation entre un objet du modèle et la source de données sous-jacente.
Objet CalculationDefinition	Agit comme un modèle pour définir des catégories calculées.
Objet Category	Définit une catégorie utilisée dans un modèle.
Objet CategorySet	Regroupe des catégories utilisées dans une définition de calcul.
Objet ChildCube	Définit un cube comme membre d'un groupe de cubes.
Objet Column	Représente une donnée élémentaire unique dans une source de données.
Objet CrossTabDataSource	Représente un fichier externe au format de tableau.
Objet Cube	Définit un cube.
Objet CubeGroup	Définit un ensemble de cubes enfants associés.
Objet CurrencyRate	Définit le taux de change de la conversion monétaire
Objet CurrencyRecord	Définit la manière dont les informations relatives aux devises apparaissent.
Objet CurrencyTable	Définit une table des devises utilisée dans la conversion monétaire.
Objet CustomView	Représente une vue personnalisée.
Objet DataSource	Définit une source de données générique.
Objet DateDimension	Organise les données de date dans des groupes logiques.

<b>Objet</b>	<b>Description</b>
Objet DateDrillDown	Définit des branches de niveaux principales et secondaires pour des valeurs de date ou de temps.
Objet DateLevel	Définit un niveau de date dans un modèle.
Objet DateWizard	Crée un objet DateDimension.
Objet DbDataSource	Représente un fichier externe au format de base de données.
Objet Dimension	Organise les données autres que de type date dans des groupes logiques.
Objet DrillDown	Définit des branches de niveaux principales et secondaires pour naviguer dans les cubes.
Objet DrillThroughTarget	Représente une liaison d'accès au détail vers un rapport externe ou un PowerCube.
Objet Filter	Représente un filtre dans le modèle Transformer.
Objet FlatFileDataSource	Représente un fichier externe au format ASCII.
Objet IqdDataSource	Représente un fichier de définition de requête d'Impromptu (.iqd).
Objet de niveau	Définit un niveau dans une dimension.
Objet Measure	Représente des valeurs quantitatives dans un cube.
Objet Model	Définit un modèle Transformer.
Objet Name	Référence une chaîne de texte qui représente un nom ou un message.
Objet Namespace	Représente un espace-noms dans le modèle Transformer.
Objet Package	Représente une source de données de pack dans le modèle Transformer.
Objet PackageDataSourceConnection	Représente une connexion de source de données de pack dans le modèle Transformer.
Objet Prompt	Représente une invite dans le modèle Transformer.

Objet	Description
Objet Query	Représente une requête faisant partie d'un rapport ou d'un pack.
Objet Report	Représente une source de données de rapport dans le modèle Transformer.
Objet SecurityObject	Représente un objet de sécurité dans le modèle Transformer.
Objet de code d'accès	Définit un code d'accès à la base de données.
Objet SpecialCategory	Définit une catégorie avec une valeur qui ne provient pas directement d'une source de données.
Objet SuspendedModel	Représente un modèle incomplet existant.
Objet View	Définit une vue partielle d'une dimension.

---

## Objet Application

L'objet Application représente une instance de l'application Transformer.

### Explication

Utilisez cet objet pour créer ou modifier un modèle.

La plupart des propriétés de l'objet Application définissent des préférences d'exécution. Une fois les propriétés de préférences définies, elles restent en vigueur jusqu'à ce que vous les changiez, à savoir que vous n'avez pas à les définir chaque fois que vous créez un objet Application.

Pour créer cet objet, utilisez la fonction CreateObject.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode NewModel	Crée un modèle vide.
Méthode OpenModel	Ouvre un modèle existant.
Méthode OpenSuspendedModel	Ouvre un modèle en attente.

Méthode	Description
Méthode RemoveSuspendedModel	Supprime un modèle en attente de la collection.

Propriété	Description
Propriété PatFile	Définit ou retourne l'emplacement du fichier de modèle cogtr_langue.pat, tel que cogtr_en.pat, pour l'environnement local du produit associé lorsque la conception automatique est utilisée.
Propriété CurrentModel	Retourne l'objet Model ouvert.
Propriété DataCharacterSet	Définit ou retourne le jeu de caractères par défaut utilisé par l'application.
Propriété DataSourcePath	Définit ou retourne l'emplacement dans lequel Transformer recherche les fichiers de la source de données.
Propriété DataTemporaryFilePath	Définit ou retourne le nom du répertoire dans lequel Transformer crée des fichiers de travail temporaires lorsqu'il génère des cubes.
Propriété DefaultDateFormat	Définit ou retourne le paramètre par défaut de la propriété DateInputFormat.
Propriété DetachDataSource	Définit ou retourne une valeur indiquant si la connexion à la source de données est maintenue ou libérée.
Propriété EnableMessageLogging	Définit ou retourne une valeur indiquant si les messages de Transformer sont consignés dans un fichier journal.
Propriété LogErrorLevel	Définit ou retourne le niveau de gravité des messages d'erreur consignés.
Propriété LogFileAppend	Définit ou retourne une valeur indiquant si Transformer ajoute des messages au fichier journal ou s'il écrase les messages antérieurs de journal.
Propriété LogFileName	Définit ou retourne le nom du fichier journal.
Propriété LogFilePath	Définit ou retourne l'emplacement dans lequel Transformer enregistre le fichier journal.

Propriété	Description
Propriété MaxTransactionNumber	Définit ou retourne le nombre maximum d'enregistrements que Transformer traite avant d'enregistrer les modifications dans un cube.
Propriété ModelsPath	Définit ou retourne l'emplacement dans lequel Transformer ouvre et enregistre les fichiers modèles.
Propriété ModelTemporaryFilePath	Définit ou renvoie l'emplacement dans lequel Transformer crée des fichiers modèles temporaires (.qy?).
Propriété Name	Renvoie le nom de l'application.
Propriété PatFile	Définit ou renvoie l'emplacement du fichier de modèle cogtr_ <i>langue</i> .pat, tel que cogtr_en.pat, pour l'environnement local du produit associé lorsque la conception automatique est utilisée.
Propriété PowerCubesPath	Définit ou renvoie l'emplacement dans lequel Transformer crée des fichiers de PowerCube (.mdc).
Propriété PowerPlayPath	Définit ou renvoie l'emplacement de PowerPlay.exe.
Propriété RowsAsSample	Définit ou renvoie le nombre de lignes que la méthode DoAutoDesign utilise comme échantillon pour créer un modèle.
Propriété RowsChecked	Définit ou renvoie le nombre maximal de lignes que la méthode DoAutoDesign lit à partir de la source de données.
Propriété ServicesBuildNumber	Renvoie le numéro de création du service d'accès aux données utilisé par Transformer.
Propriété ServicesVersionText	Renvoie le numéro de version du service d'accès aux données utilisé par Transformer.
Propriété SortComparisonRule	Définit ou renvoie la règle de comparaison utilisée par Transformer pour trier des données.
Propriété SuspendedModels	Renvoie une collection d'objets de modèles en attente.
Propriété TransdaPath	Définit ou renvoie l'emplacement de Transda.exe.
Propriété Type	Renvoie le type de l'objet.

Propriété	Description
Propriété Version	Renvoie le numéro de version de Transformer.

## Exemples

```
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
```

---

## Objet Association

L'objet Association définit la relation entre un objet du modèle et la source de données sous-jacente.

### Explication

Cet objet permet de gérer la liaison entre des objets d'un modèle et leurs sources de données sous-jacentes.

Chaque objet Association d'une collection possède un rôle. Une association peut, par exemple, référencer des valeurs sources, afficher des noms abrégés ou définir un ordre de tri. Pour spécifier le rôle d'un objet Association, utilisez la propriété AssociationRole.

Chaque objet Association référence une colonne de la source de données. L'utilisation de la colonne par l'objet Association dépend du rôle qui lui est affecté. Si vous souhaitez spécifier une colonne, définissez la propriété Libellé sur le nom de l'objet de référence.

Pour créer un objet Association, utilisez la méthode Add de la collection Associations ou encore les méthodes AssociateWith ou DimensionAssociateWith d'un objet applicable.

Par défaut, des catégories apparaissent dans un niveau, puis dans un cube selon leur ordre d'apparition dans la source de données. Pour trier ces objets, utilisez un objet Association et affectez à la propriété AssociationRole la valeur trAssociationOrderBy.

Quand Transformer trie un niveau, il base le tri sur la colonne spécifiée dans la propriété Label. Vous pouvez, par exemple, trier un niveau Produit par nom de produit ou numéro de produit.

Utilisez la propriété Context pour nommer la branche de niveaux à laquelle s'applique le tri. Quand un niveau représente la convergence de deux branches de niveaux ou davantage, vous pouvez appliquer un ordre de tri différent à chacune d'elles.

Utilisez la propriété OrderByDescending pour spécifier le classement du tri. Si vous définissez le tri à l'aide d'une colonne de données numériques, spécifiez la taille du type de données à l'aide de OrderByStorageType.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.



## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Association.
Méthode Update	Met à jour l'objet Association.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété AssociationRole	Définit ou retourne le rôle de l'objet Association.
Propriété AssociationType	Définit ou retourne le type de la source de données associée à un objet Association.
Propriété Context	Définit ou retourne la branche de niveaux utilisée pour classer des catégories au sein d'un niveau.
Propriété Label	Définit ou retourne le nom de l'objet de référence.
Propriété Parent	Retourne l'objet parent.
Propriété RefreshDescription	Définit une colonne de la source de données pour l'objet Association.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()
```

---

## Objet CalculationDefinition

L'objet CalculationDefinition agit comme un modèle pour définir des catégories calculées.

### Explication

Les objets CalculationDefinition permettent d'ajouter des expressions complexes qui incluent les valeurs d'une ou de plusieurs catégories dans le calcul.

Pour créer un objet CalculationDefinition, utilisez la méthode Add de la collection CalculationDefinitions.

La valeur d'un objet CalculationDefinition est basée sur une formule ainsi que sur les catégories ou les ensembles de catégories auxquels s'applique la formule. Vous composez la formule sous forme d'expression à l'aide de fonctions et d'opérateurs, puis vous l'affectez à la propriété ExpressionText. L'expression peut référencer plusieurs ensembles de catégories.

Les catégories décrites par l'expression sont générées quand vous exécutez la méthode Update. La validité de l'expression est aussi vérifiée. En présence d'erreurs, telles que des catégories en double, la mise à jour échoue et Transformer génère des messages d'erreur.

La propriété Description, bien qu'elle soit disponible en tant que propriété, n'a aucun effet. Toute modification apportée à la propriété Description ne sera pas conservée.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CalculationDefinition.
Méthode Update	Met à jour l'objet CalculationDefinition.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété CategorySets	Retourne la collection CategorySets utilisée par l'objet CalculationDefinition.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur de l'objet CalculationDefinition.
Propriété Group	Définit ou retourne une valeur indiquant si l'objet CalculationDefinition est regroupé avec les catégories de l'ensemble de catégories.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
calcDef = dimension.CalculationDefinitions.Add()
```

---

## Objet Category

L'objet Category définit une catégorie utilisée dans un modèle.

### Explication

Cet objet permet de modifier les propriétés d'un objet Category dans un modèle. Les catégories sont générées à partir de la source de données quand vous utilisez la méthode GenerateCategories. Elles sont aussi générées quand vous créez un cube.

Une catégorie ordinaire contient des données structurales qui décrivent ou classent les détails d'une organisation, tels qu'une gamme de produits ou le site du vendeur. Les catégories sont généralement associées à des niveaux au sein d'une dimension. Quand vous créez un modèle, les catégories sont complétées à l'aide des valeurs de mesure des sources de données transactionnelles.

Une catégorie calculée utilise des expressions et des valeurs catégorielles pour dériver sa propre valeur. Pour créer une catégorie calculée, utilisez la méthode `Add` de la collection `Categories`, puis affectez-lui une valeur à l'aide de la propriété `ExpressionText`. Vous pouvez aussi utiliser des objets `CalculationDefinition` et `CategorySet` pour créer des catégories calculées.

Pour retourner la collection du niveau supérieur des catégories ordinaires d'une dimension, utilisez la propriété `Categories` de l'objet `DrillDown`. Pour retourner une collection de catégories enfants, utilisez la propriété `ChildCategories` de l'objet `Category`. Pour retourner une collection de catégories spéciales, utilisez la propriété `Categories` de la dimension parent.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode `Update`.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode <code>ConnectWithCategory</code>	Déplace une catégorie enfant vers une nouvelle catégorie parent dans le même niveau.
Méthode <code>Delete</code>	Supprime l'objet <code>Category</code> .
Méthode <code>MoveToCategory</code>	Déplace une catégorie enfant vers une autre catégorie parent.
Méthode <code>MoveToLevel</code>	Déplace une catégorie enfant vers un nouveau niveau de la catégorie parent en cours.
Méthode <code>SetAllocation</code>	Modifie le type d'allocation de la mesure utilisée par l'objet.
Méthode <code>Update</code>	Met à jour l'objet <code>Category</code> .

Propriété	Description
Propriété <code>AllocationMeasure</code>	Retourne l'objet <code>Measure</code> utilisé comme facteur de pondération.
Propriété <code>AllocationType</code>	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété <code>Application</code>	Retourne l'objet <code>Application Transformer</code> .

Propriété	Description
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété ChildCategories	Retourne une collection Categories.
Propriété Code	Définit ou retourne le code d'identification unique d'une catégorie au sein d'une dimension.
Propriété Description	Définit ou retourne la description de l'objet Category.
Propriété Dimension	Retourne la dimension d'un objet Category.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur de l'objet Category.
Propriété Format	Définit ou retourne une valeur indiquant la manière dont les valeurs numériques apparaissent.
Propriété FormatDecimals	Définit ou retourne le nombre de décimales que PowerPlay affiche pour la mesure.
Propriété HideValue	Spécifie si la valeur d'un objet Category doit être masquée. Valeur par défaut : false.
Propriété Inclusion	Définit ou retourne une valeur indiquant à quel moment une catégorie est incluse dans un cube.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété IsPrimary	Retourne une valeur indiquant si l'objet Category est la catégorie de la branche principale ou secondaire.
Propriété KeyName	Définit ou retourne la valeur qui apparaît dans la colonne associée de la source de données.

Propriété	Description
Propriété Label	Définit ou retourne le nom de l'objet. Dans l'interface utilisateur de Transformer, le libellé est "Valeur source", mais la valeur par défaut est "".
Propriété LastUseDate	Retourne la date de la dernière modification ou de la dernière utilisation de l'objet Category.
Propriété Level	Retourne le niveau d'un objet Category.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Orphanage	Définit ou retourne une valeur indiquant si une catégorie est isolée.
Propriété Parent	Retourne l'objet parent.
Propriété ParentCategories	Retourne une collection ParentCategories.
Propriété Partition	Définit ou retourne le numéro d'une partition manuelle.
Propriété ReverseSign	Définit ou retourne une valeur indiquant si PowerPlay inverse le signe d'une mesure.
Propriété ShortName	Définit ou retourne le nom abrégé de l'objet Category.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
parentCategory = drillDown.Categories.Item(1)'select
"Camping Equipment"
```

---

## Objet CategorySet

L'objet CategorySet regroupe des catégories utilisées dans une définition de calcul.

### Explication

Certaines expressions d'un objet CalculationDefinition peuvent prendre un objet CategorySet comme paramètre. Un ensemble de catégories contient une ou plusieurs catégories d'une dimension applicable qui fournit les valeurs de l'expression.

Vous devez créer l'ensemble de catégories avant de le référencer dans une expression.

Un ensemble de catégories constitue une méthode pratique pour regrouper des catégories lorsque vous ne voulez pas inclure la totalité du niveau dans votre formule. Un ensemble de catégories peut aussi contenir des catégories de niveaux différents dans une dimension.

Commencez par créer un objet CalculationDefinition, puis utilisez la méthode Add pour ajouter un objet CategorySet à la collection CategorySets. Utilisez ensuite la méthode Add pour ajouter des catégories à la collection Categories de l'objet CategorySet. Pour terminer, nommez l'ensemble de catégories à l'aide de la propriété Label.

Utilisez la propriété Label de l'objet CategorySet comme référence dans l'expression. Veillez à placer entre des guillemets supplémentaires la chaîne de code et le libellé de chaque catégorie.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CategorySet.
Méthode Update	Met à jour l'objet CategorySet.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Categories	Retourne la collection des catégories qui sont incluses dans l'objet CategorySet.
Propriété Description	Définit ou retourne la description de l'objet CategorySet.
Propriété Label	Définit ou retourne un nom descriptif qui apparaît dans PowerPlay.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
catSet = calcDef.CategorySets.Add()
```

---

## Objet ChildCube

L'objet ChildCube définit un cube comme membre d'un groupe de cubes.

### Explication

Un groupe de cubes représente un niveau dans une dimension. Chaque objet ChildCube du groupe reflète une catégorie unique au sein de ce niveau.

Cet objet permet de modifier les propriétés d'un objet ChildCube. Pour retourner un objet ChildCube à partir d'une collection, utilisez la propriété ChildCubes de l'objet CubeGroup.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode RemoveCubeCustomView	Supprime la vue personnalisée.
Méthode Update	Met à jour l'objet ChildCube.

Propriété	Description
Propriété AllowDrillThrough	Définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.
Propriété AltMDCFile	Définit ou renvoie un autre nom de fichier pour le cube enfant.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Consolidate	Définit ou retourne une valeur indiquant le mode de consolidation d'un cube.
Propriété CubeCreation	Définit ou retourne une valeur indiquant si le cube est créé.
Propriété CubeCustomViews	Retourne une collection d'objets CustomView.
Propriété Description	Définit ou retourne la description de l'objet ChildCube.
Propriété DrillThroughTargets	Renvoie la collection de tous les objets DrillThroughTargets associés à un objet ChildCube.

Propriété	Description
Propriété IsMDCInUse	Retourne une valeur qui indique si un cube enfant est en cours d'utilisation ou de construction.
Propriété MDCFile	Définit ou retourne le nom d'un fichier de PowerCube (.mdc).
Propriété MeasureName	Définit ou retourne un titre descriptif qui identifie une mesure sur la ligne des dimensions de PowerPlay.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Optimize	Définit ou retourne l'option d'optimisation du cube en cours.
Propriété Parent	Retourne l'objet parent.
Propriété Password	Définit un mot de passe pour le cube enfant (prenant en compte la distinction majuscule/minuscule).
Propriété Status	Retourne le statut d'un problème associé au cube lors de la dernière création.
Propriété Type	Retourne le type de l'objet.
Propriété UseAltMDCFile	Définit ou retourne une valeur qui indique si un nom de fichier temporaire peut être utilisé.

## Exemples

```
objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)objChildCube =
objCubesByRegion.ChildCubes("Central Europe")
```

---

## Objet Column

L'objet Column représente une donnée élémentaire unique dans une source de données.

### Explication

Les objets Column du modèle sont liés à des colonnes de la source de données qui leur est associée et se comportent comme la source des valeurs des niveaux et des mesures une fois que des catégories ont été générées.

Transformer utilise des colonnes pour construire des dimensions et des mesures dans le modèle. Une colonne contient soit des valeurs de texte qui deviennent des catégories dans le modèle, soit des valeurs numériques qui deviennent des valeurs de mesure. Une colonne peut aussi contenir des valeurs utilisées sous la forme de libellés, de noms abrégés ou de descriptions textuelles des catégories.



Pour la plupart des types de données sources pris en charge, Transformer identifie automatiquement les colonnes des sources de données et affecte des propriétés ainsi que des noms de colonne par défaut. Vous pouvez modifier ces affectations ultérieurement, si vous le souhaitez. Si vous créez des colonnes manuellement, lors de l'utilisation d'un fichier de texte à zones fixes, par exemple, il suffit d'identifier les colonnes de données à utiliser dans le modèle.

Quand un même nom de colonne apparaît dans deux sources de données ou davantage, Transformer associe ces colonnes à un niveau ou à une mesure unique. Il n'utilise pas de colonnes correspondantes pour effectuer des jointures aux fichiers source, mais fait plutôt appel à chaque fichier source pour compléter le modèle à l'aide de catégories.

Pour inclure une dimension de temps dans votre modèle, au moins une source de données du modèle doit contenir une colonne qui représente des valeurs de date.

Si votre modèle nécessite plus d'informations structurelles ou numériques que votre source de données n'en contient, vous pouvez définir une colonne calculée à l'aide de la propriété ExpressionText. Une colonne calculée utilise d'autres colonnes, fonctions et constantes pour dériver des nouvelles données pour le modèle.

Pour créer un objet Column, utilisez la méthode Add de la collection Columns.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Column.
Méthode Update	Met à jour l'objet Column.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété DataClass	Définit ou retourne le type de données d'une colonne source.
Propriété DateDegreeofDetail	Définit ou retourne le niveau de date auquel des rapports sont générés.
Propriété DateDegreeofDetailLevelName	Définit ou retourne le niveau de date qui s'applique à une mesure cumule extérieurement.
Propriété DateInputFormat	Définit ou retourne le format d'affichage des dates.

<b>Propriété</b>	<b>Description</b>
Propriété Decimals	Retourne le nombre de décimales s'il est défini dans les données sources.
Propriété Description	Définit ou retourne la description de l'objet Column.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur d'une colonne calculée.
Propriété InputScale	Définit ou retourne la valeur d'échelle utilisée pour convertir les nombres décimaux des colonnes en entiers.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Origin	Définit ou retourne la source de l'objet Column.
Propriété OriginalName	Définit ou retourne le nom de la colonne dans la source de données.
Propriété OutputScale	Définit ou retourne la valeur d'échelle utilisée pour convertir les nombres entiers en nombres décimaux dans PowerPlay.
Propriété Parent	Retourne l'objet parent.
Propriété Position	Définit ou retourne le nombre ordinal ou la position de départ de la colonne dans la source de données.
Propriété Size	Spécifie la taille de la colonne pour certaines sources de données.
Propriété StorageType	Définit ou retourne le type de stockage numérique de l'objet, lorsqu'il est applicable.
Propriété TimeArrayColumn	Définit ou retourne le nom de la première colonne du tableau quand l'objet représente un tableau de dates.
Propriété TimeArrayStartMonth	Définit ou retourne le mois auquel commence l'exercice fiscal quand l'objet inclut un tableau de dates.
Propriété TimeArrayType	Définit ou retourne le type de tableau utilisé pour les valeurs de date.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objDataSource  
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objColumn  
= objDataSource.Columns("EXPECTED_VOLUME")
```

---

## Objet CrossTabDataSource

L'objet Cross TabDataSource représente un fichier externe au format de tableau croisé.

### Explication

Un objet CrossTabDataSource référence une feuille de calcul Lotus 1-2-3 ou Microsoft Excel. Pour créer une source de données de tableau, affectez un nom de plage à chaque colonne de la feuille de calcul.

Pour créer un objet CrossTabDataSource, utilisez la méthode Add de la collection DataSources. Spécifiez le paramètre trCrossTabDataSource à partir de la liste des valeurs de xtrObjectType.

Affectez à la propriété SourceType la valeur trExcelCrossTab ou trLotus123CrossTab.

Quand Transformer ouvre une source de données de tableau, il crée une collection de colonnes dans le modèle, à savoir une colonne pour chaque plage nommée. Transformer détermine la classe de données de chaque colonne (texte, nombre ou date) sur la base d'un échantillon des valeurs des cellules dans chaque plage.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CrossTabDataSource.
Méthode Move	Déplace l'objet CrossTabDataSource à une autre position dans la collection DataSources.
Méthode Update	Met à jour l'objet CrossTabDataSource.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Columns	Retourne une collection Columns.

<b>Propriété</b>	<b>Description</b>
Propriété ColumnsLoaded	Retourne une valeur indiquant si les colonnes d'une source de données ont été utilisées pour créer un modèle.
Propriété Description	Définit ou retourne la description de l'objet CrossTabDataSource.
Propriété DecimalPoint	Définit ou retourne le caractère de séparation utilisé comme virgule décimale.
Propriété External	Définit ou retourne une valeur indiquant si la source de données contient des valeurs récapitulatives.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété LocalPath	Définit ou retourne l'emplacement de la source de données locale.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom unique de la source de données.
Propriété Parent	Retourne l'objet parent.
Propriété ServerQuery	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété SetsCurrentPeriod	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété ThousandPoint	Définit ou retourne le caractère qui sépare les milliers.
Propriété SourceType	Retourne le type de l'objet.

---

## Objet Cube

L'objet Cube définit un cube.

### Explication

Cet objet permet de modifier les propriétés d'un cube. Chaque cube défini dans un modèle fournit aux utilisateurs une vue multidimensionnelle des données sous-jacentes par l'intermédiaire de PowerPlay. Pour créer des objets Cube, utilisez la méthode Add de la collection Cubes.

Utilisez la méthode CreateMDCFile pour créer le fichier de PowerCube (.mdc). Utilisez la méthode CreateMDCFiles de l'objet Model pour créer tous les fichiers .mdc du modèle.

Les utilisateurs ne doivent se connecter qu'une seule fois au cours de la même session pour afficher un nombre quelconque de cubes et de rapports associés à leur vue personnalisée.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

Les méthodes PublishDatasource et méthode PublishPackage permettent aux utilisateurs de publier une source de données ou un pack. Lors de la publication d'un pack, la source de données doit également être publiée par défaut. Au moment de la publication, les utilisateurs peuvent choisir de remplacer ou non la source de données ou le pack. Ils peuvent affecter des vues personnalisées à chaque cube. Les vues personnalisées contiennent des informations relatives à la sécurité.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AddDeployLocation	Ajoute un emplacement de déploiement pour la copie et l'activation.
Méthode ClearDeployLocations	Supprime tous les emplacements de déploiement.
Méthode CreateMDCFile	Crée un fichier de PowerCube (.mdc) basé sur la définition du cube.
Méthode Delete	Supprime l'objet Cube.
Méthode DeployCubes	Déploie un PowerCube vers tous les emplacements de déploiement.
Méthode PublishDatasource	Publie la connexion à la source de données d'un PowerCube vers le serveur IBM Cognos BI.

Méthode	Description
Méthode PublishPackage	Publie le pack du PowerCube sur le serveur IBM Cognos BI.
Méthode RemoveCubeCustomView	Supprime la vue personnalisée du cube.
Méthode SetDeployType	Définit le type de déploiement pour la copie et l'activation.
Méthode Update	Met à jour l'objet Cube.

Propriété	Description
Propriété AllowDrillThrough	Définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.
Propriété AlternateQueryPath	Définit ou retourne un chemin d'accès secondaire de la source de données utilisé par le cube.  Lors de l'utilisation d'une propriété AlternateQueryPath, vous devez fournir le nom de la source de données et le chemin d'accès à celle-ci.
Propriété AltMDCFile	Définit ou retourne l'autre nom de fichier pour le cube.
Propriété Application	Retourne l'objet Application Transformer.
Propriété BlockParentTotals	Retourne une valeur refusée pour les parents d'enfants exclus.
Propriété CacheCrossTabs	Définit ou retourne une valeur indiquant si les récapitulatifs sont stockés dans le cache pour le tableau PowerPlay initial du cube.
Propriété CompressMDC	Définit ou retourne une valeur indiquant si le cube est compressé pour économiser l'espace de stockage.
Propriété Consolidate	Définit ou retourne une valeur indiquant le mode de consolidation d'un cube.
Propriété CubeCreation	Définit ou retourne une valeur indiquant si le cube est créé.
Propriété CubeCustomViews	Description de la propriété CubeCustomViews.
Propriété CubeStamp	Retourne la date et l'heure de création du cube.

Propriété	Description
Propriété DataSourceWindowsLocation	Définit ou renvoie l'emplacement Windows du PowerCube sur le serveur IBM Cognos BI.
Propriété Description	Définit ou retourne la description de l'objet Cube.
Propriété DesiredPartitionSize	Définit ou retourne la taille de partition souhaitée lorsque le partitionnement automatique est activé.
Propriété DimensionView	Définit ou retourne l'objet View associé à un cube ou à une vue personnalisée.
Propriété DimensionViewType	Définit ou retourne les dimensions et les vues qui appartiennent à un cube.
Propriété DrillThroughTargets	Renvoie une collection d'objets DrillThroughTarget associés à un objet Cube.
Propriété EstimatedRows	Définit ou retourne une estimation du nombre d'enregistrements contenus dans le cube avant le découpage automatique en partitions.
Propriété IncrementalUpdate	Définit ou retourne une valeur indiquant si la mise à jour incrémentielle du cube est appliquée à partir de la source de données.
Propriété IsMDCInUse	Retourne une valeur qui indique si un cube enfant est en cours d'utilisation ou de construction.
Propriété MaxNumPartLevels	Définit ou retourne le nombre maximum de fois que Transformer lit la source de données pendant le découpage en partitions du cube.
Propriété MDCFile	Définit ou retourne le nom d'un fichier de PowerCube (.mdc).
Propriété MeasureInclude	Définit ou retourne le nom d'une mesure à inclure dans l'objet Cube.
Propriété MeasureName	Définit ou retourne un titre descriptif qui identifie une mesure sur la ligne des dimensions de PowerPlay.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Optimize	Définit ou retourne l'option d'optimisation du cube en cours.
Propriété Parent	Retourne l'objet parent.

Propriété	Description
Propriété Password	Définit un mot de passe pour le cube (prenant en compte la distinction majuscule/minuscule).
Propriété Server	Définit ou retourne une valeur indiquant si un cube est traité localement ou sur un serveur.
Propriété Signon	Définit ou retourne l'objet de code d'accès utilisé par le cube.
Propriété Status	Retourne le statut d'un problème associé au cube lors de la dernière création.
Propriété Type	Retourne le type de l'objet.
Propriété UseAltMDCFile	Définit ou retourne une valeur qui indique si un nom de fichier temporaire peut être utilisé.

## Exemples

```
objCube = objModel.Cubes.Item(1)
```

---

## Objet CubeGroup

L'objet CubeGroup définit un ensemble de cubes enfants associés.

### Explication

Cet objet permet de définir le contenu d'un groupe de cubes. Un groupe de cubes représente un niveau dans une dimension. Chaque objet ChildCube du groupe reflète une catégorie unique au sein de ce niveau.

Pour créer un objet CubeGroup, utilisez la méthode Add de la collection Cubes et affectez au paramètre Type la valeur trCubeGroup.

Utilisez les propriétés GroupDimension et GroupLevel pour déterminer le contenu du groupe de cubes. Les fichiers de PowerCube (.mdc) proprement dits sont créés à l'aide de la méthode CreateMDCFile.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode CreateMDCFile	Crée des fichiers de PowerCube (.mdc) basés sur la définition du groupe de cubes.



<b>Méthode</b>	<b>Description</b>
Méthode Delete	Supprime l'objet CubeGroup et tous les cubes enfants.
Méthode Update	Met à jour l'objet CubeGroup.

<b>Propriété</b>	<b>Description</b>
Propriété AllowDrillThrough	Définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.
Propriété AlternateQueryPath	Définit ou retourne une source de données alternative utilisée par des cubes du groupe.
Propriété AltMDCFile	Définit ou retourne l'autre nom de fichier pour le cube.
Propriété Application	Retourne l'objet Application Transformer.
Propriété CacheCrossTabs	Définit ou retourne une valeur indiquant si les récapitulatifs sont stockés dans le cache pour le tableau PowerPlay initial d'un groupe de cubes.
Propriété ChildCubes	Retourne une collection d'objets ChildCube à partir d'un groupe de cubes.
Propriété CompressMDC	Définit ou retourne une valeur indiquant si les cubes sont compressés pour économiser l'espace de stockage.
Propriété Consolidate	Définit ou retourne une valeur indiquant le mode de consolidation d'un cube.
Propriété CubeCreation	Définit ou retourne une valeur indiquant si le cube est créé.
Propriété CubeCustomViews	Description de la propriété CubeCustomViews.
Propriété CubeStamp	Retourne la date et l'heure de création du cube.
Propriété Description	Définit ou retourne la description de l'objet CubeGroup.
Propriété DesiredPartitionSize	Définit ou retourne la taille souhaitée pour chaque partition des cubes quand le découpage automatique en partitions est activé.

Propriété	Description
Propriété DetailLevel	Définit ou retourne le niveau de détail le plus bas des cubes dans un objet CubeGroup.
Propriété DimensionView	Définit ou retourne l'objet View associé à un cube ou à une vue personnalisée.
Propriété DimensionViewType	Définit ou retourne les dimensions et les vues qui appartiennent à un groupe de cubes.
Propriété DrillThroughTargets	Renvoie une collection d'objets DrillThroughTarget associés à un objet CubeGroup.
Propriété EstimatedRows	Définit ou retourne une estimation du nombre d'enregistrements contenus dans le cube avant le découpage automatique en partitions.
Propriété GroupDimension	Définit ou retourne la dimension utilisée pour créer le groupe de cubes.
Propriété GroupLevel	Définit ou retourne le niveau auquel appartiennent les catégories qui deviennent les cubes individuels du groupe de cubes.
Propriété IncrementalUpdate	Définit ou retourne une valeur indiquant si la mise à jour incrémentielle du groupe de cubes est appliquée à partir de la source de données.
Propriété IsMDCInUse	Retourne une valeur qui indique si un cube enfant est en cours d'utilisation ou de construction.
Propriété IsTimeBasedPartitionedCube	Définit ou retourne une valeur qui indique si un cube est défini comme cube découpé en partitions de temps.
Propriété MaxNumPartLevels	Définit ou retourne le nombre maximum de fois que Transformer lit la source de données pendant le découpage en partitions du cube.
Propriété MDCFile	Définit ou retourne le nom d'un fichier de PowerCube (.mdc).
Propriété MeasureInclude	Définit ou retourne le nom d'une mesure à inclure dans le groupe de cubes.
Propriété MeasureName	Définit ou retourne un titre descriptif qui identifie une mesure sur la ligne des dimensions de PowerPlay.

Propriété	Description
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Optimize	Définit ou retourne l'option d'optimisation du cube en cours.
Propriété Parent	Retourne l'objet parent.
Propriété Password	Définit un mot de passe pour le groupe de cubes (prenant en compte la distinction majuscule/minuscule).
Propriété Server	Définit ou retourne une valeur indiquant si un cube est traité localement ou sur un serveur.
Propriété Signon	Définit ou retourne l'objet de code d'accès utilisé par le groupe de cubes.
Propriété Status	Retourne le statut d'un problème associé au cube lors de la dernière création.
Propriété SummaryLevel	Définit ou retourne le niveau à utiliser pour récapituler des catégories externes dans le groupe de cubes.
Propriété Type	Retourne le type de l'objet.
Propriété UseAltMDCFile	Définit ou retourne une valeur qui indique si un nom de fichier temporaire peut être utilisé.

## Exemples

```
objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)
```

---

## Objet CurrencyRate

L'objet CurrencyRate définit le taux de change appliqué pour la conversion monétaire.

### Explication

Les objets CurrencyRate sont gérés manuellement ou complétés par une source de données. Chaque taux de conversion est associé à une catégorie de date qui définit la période de validité d'un taux donné.

Des objets CurrencyRate sont automatiquement ajoutés à un objet CurrencyRecord. Le nombre d'objets associés à un objet CurrencyRecord correspond au niveau de date spécifié par la propriété DateLevel, ainsi que par la plage de dates définie dans la dimension de temps. Quand le niveau de détail est le mois et que la plage de dates s'étale sur deux ans, par exemple, un enregistrement de devise contient 24 taux de conversion.

Pour que la conversion monétaire fonctionne, il doit y avoir des objets CurrencyRate, CurrencyRecord, CurrencyTable et DateDimension.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CurrencyRate.
Méthode Update	Met à jour l'objet CurrencyRate.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Category	Retourne la catégorie de date qui détermine la validité du taux de conversion.
Propriété CurrencyRecord	Retourne l'objet CurrencyRecord auquel s'applique l'objet CurrencyRate.
Propriété CurrencyTable	Retourne l'objet CurrencyTable associé.
Propriété Parent	Retourne l'objet parent.
Propriété PopulateByDataSource	Définit ou retourne une valeur indiquant si le taux de conversion est obtenu à partir d'une source de données ou s'il est indiqué dans Transformer.
Propriété Rate	Définit ou retourne le taux de change de la devise.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRate  
= objCurrencyRecord.CurrencyRates(intX)
```

---

## Objet CurrencyRecord

L'objet Currency Record définit la manière dont les informations monétaires sont affichées.

### Explication

Cet objet permet de :

- spécifier le niveau de date d'un taux de conversion,
- définir les informations requises pour afficher une devise dans PowerPlay

Ces informations incluent le nom, le code, le symbole et les décimales de la devise. Les propriétés CurrencyDecimals et CurrencySymbol ne peuvent être modifiées que si vous affectez la valeur True à la propriété CurrencyFormatOverride.

Quand vous remplissez une table des devises à l'aide d'une source de données, des objets CurrencyRecord sont automatiquement ajoutés au modèle, ainsi que les taux de conversion associés à chaque enregistrement.

Pour créer explicitement un objet CurrencyRecord, utilisez la méthode Add de la collection CurrencyRecords. Dans ce cas, bien que des objets CurrencyRate soient générés automatiquement, vous devez gérer manuellement les taux.

Pour que la conversion monétaire fonctionne, il doit y avoir des objets CurrencyRate, CurrencyRecord, CurrencyTable et DateDimension.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CurrencyRecord.
Méthode Update	Met à jour l'objet CurrencyRecord.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété CountryCode	Définit ou retourne le code ISO du pays ou de la région auxquels s'applique l'enregistrement de devise.
Propriété CurrencyCountryLabel	Définit ou retourne le libellé du pays ou de la région de la devise.
Propriété CurrencyDecimals	Définit ou retourne le nombre de décimales utilisées dans la devise.
Propriété CurrencyFormatOverride	Définit ou retourne une valeur indiquant si vous pouvez écraser le format standard d'une devise.
Propriété CurrencyIsEMU	Définit ou retourne une valeur indiquant si l'enregistrement est un enregistrement de devise de l'UEM.
Propriété CurrencyIsEuro	Définit ou retourne une valeur indiquant si l'enregistrement est la devise euro de base.
Propriété CurrencyRates	Retourne la collection CurrencyRates.
Propriété CurrencySymbol	Définit ou retourne le symbole monétaire associé à la devise.

Propriété	Description
Propriété DateLevel	Définit ou retourne le niveau d'une dimension de temps auquel s'applique les taux de conversion.
Propriété EMUEntryDate	Définit ou retourne la date à laquelle commencent les calculs de triangulation de l'euro.
Propriété Label	Définit ou retourne un nom descriptif qui apparaît dans PowerPlay.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()
```

---

## Objet CurrencyTable

L'objet CurrencyTable définit une table des devises utilisée pour la conversion monétaire.

### Explication

Cet objet permet de définir des enregistrements de devise et des taux de conversion.

Pour créer un objet CurrencyTable, utilisez la méthode Add de la collection CurrencyTables. Quand vous remplissez une table des devises à l'aide d'une source de données, des objets CurrencyRecord sont automatiquement ajoutés au modèle, ainsi que les taux de conversion associés à chaque enregistrement.

Vous pouvez aussi ajouter manuellement des enregistrements de devise à une table des devises. Dans ce cas, des objets CurrencyRate sont générés automatiquement pour chaque nouvel objet CurrencyRecord, mais vous devez gérer manuellement les taux.

Pour que la conversion monétaire fonctionne, il doit y avoir des objets CurrencyRate, CurrencyRecord, CurrencyTable et DateDimension.

Pour convertir des devises qui n'appartiennent pas à l'Union économique et monétaire (UEM) ainsi que des devises de l'UEM antérieures à 1999, une seule table est requise et la propriété CurrencyTableType doit avoir la valeur trCurrencyTableBase. La table de base affiche initialement la devise par défaut, telle que le dollar américain. Celle-ci varie en fonction des paramètres régionaux de votre système d'exploitation. Vous pouvez remplacer ce paramètre par n'importe quelle autre devise, y compris l'euro.

Les conversions de devises de l'UEM ultérieures à 1998 exigent deux tables. Affectez la valeur trCurrencyTableBase à la propriété CurrencyTableType d'une table et la valeur trCurrencyTableEuro à la propriété de l'autre table. La table de

l'euro contient les taux de conversion fixes utilisés pour convertir des valeurs vers ou depuis les devises nationales des pays ou régions de l'Union économique et monétaire en utilisant l'euro et les devises de base comme valeurs intermédiaires de la conversion. Ce processus porte le nom de triangulation de l'euro. Pour que la triangulation de l'euro fonctionne, trie les niveaux de date dans l'ordre croissant et affectez la valeur True à la propriété Unique de chaque niveau de date. Affectez la valeur zéro (0) à la zone de date des enregistrements des membres de l'Union économique et monétaire postérieurs au 1er janvier 1999.

Les multiplications, divisions et opérations d'arrondissement requises sont effectuées dans PowerPlay.

Vous pouvez utiliser plusieurs sources de données pour fournir les données relatives aux devises dans votre table de base. Cependant, tous les noms de colonne doivent correspondre afin que les informations se combinent correctement. Cela s'applique aussi à la table de l'euro, même si vous devez affecter des noms distincts aux colonnes associées à celle-ci.

Quand vous complétez une table à l'aide d'une source de données, affectez la valeur `trGenerationNoCreatePowerCubes` à la propriété `GeneratePowerCube` et la valeur `False` à la propriété `GenerateCategories`. Cela permet d'éviter toute interférence entre l'objet `CurrencyTable` et les catégories générées dans les autres parties de votre modèle. Ces propriétés peuvent demeurer activées si les noms des colonnes de date de la source de données de base et de la source de données de l'euro diffèrent de ceux de votre dimension de temps et entre eux. Si tel n'était pas le cas, `Transformer` ne pourrait pas les différencier.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode `Update`.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode <code>AssociateWith</code>	Associe un objet source à un objet <code>CurrencyTable</code> .
Méthode <code>Delete</code>	Supprime l'objet <code>CurrencyTable</code> .
Méthode <code>Update</code>	Met à jour l'objet <code>CurrencyTable</code> .

Propriété	Description
Propriété <code>Application</code>	Retourne l'objet <code>Application Transformer</code> .
Propriété <code>Associations</code>	Retourne une collection <code>Associations</code> .
Propriété <code>CurrencyTableType</code>	Définit ou retourne le type de la table des devises.
Propriété <code>Name</code>	Définit ou retourne le nom de l'objet.

Propriété	Description
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objBaseTable = objModel.CurrencyTables.Add()
```

---

## Objet CustomView

L'objet CustomView représente une vue personnalisée.

### Explication

Cet objet permet de fournir la sécurité de base aux cubes générés à partir d'un modèle. Chaque vue personnalisée est constituée d'objets de sécurité tels que des utilisateurs, des groupes et des rôles, qui exécutent des fonctions de tâche similaires et nécessitent donc des informations similaires.

Chaque objet CustomView intègre automatiquement une collection d'objets View, à savoir un pour chaque dimension. La vue par défaut de chaque objet d'une collection est la vue "Toutes catégories". Vous pouvez modifier cette vue à l'aide de la propriété DimensionInclude en omettant toutes les catégories ou en créant une vue personnalisée.

Les vues personnalisées fournissent aux utilisateurs de PowerPlay un sous-ensemble spécifique d'informations relatives au cube. Une vue personnalisée peut notamment résumer un niveau entier ou des catégories individuelles du niveau. Si vous spécifiez une vue personnalisée (trViewTypeCustom), vous devez aussi utiliser la propriété DimensionView pour retourner l'objet View associé à une dimension. Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété Apex ou associer l'objet View aux catégories ou aux niveaux sélectionnés au moyen de la méthode SetViewStatus.

Pour terminer une vue avec une vue personnalisée, vous devez associer cette dernière à un cube. Utilisez la méthode Add pour ajouter un objet CustomView à la collection CubeCustomViews du cube.

Les objets CustomView sont définis dans le modèle et retournés par la propriété CustomViews. Vous pouvez modifier les propriétés de l'objet CustomView de Transformer, telles que DimensionInclude, pour orienter la portée de la vue personnalisée à l'intérieur de Transformer.

Pour contrôler l'accès à la totalité ou à des parties d'un cube, définissez des groupes d'utilisateurs dans un espace-noms, puis associez les vues personnalisées à ce cube dans votre modèle Transformer.

Ouvrez, par exemple, un modèle qui contient des données de performances globales par gamme de produits, vendeur et région. Créez un groupe de cubes dans lequel chaque cube contient des données de performances détaillées pour une région et des données récapitulatives pour toutes les autres régions. Les vues personnalisées sont définies et appliquées de façon à ce que chaque équipe de vendeurs possède les informations détaillées qui concernent sa propre région.



Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet CustomView.
Méthode Update	Met à jour l'objet CustomView.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété ChildCustomViews	Retourne une collection d'objets CustomView.
Propriété Description	Retourne la description de l'objet CustomView.
Propriété DimensionInclude	Définit ou retourne le type de vue personnalisée.
Propriété DimensionView	Retourne l'objet View associé à une vue personnalisée.
Propriété MeasureInclude	Définit ou retourne une valeur booléenne afin de déterminer si une mesure s'applique à un objet CustomView.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété SecurityObjects	Retourne la collection d'objets SecurityObjects associée à la vue personnalisée.
Propriété Type	Retourne le type de l'objet.
Propriété Views	Retourne une collection Views.

## Exemples

```
custom_view = model.CustomViews.Add()
```

---

## Objet DataSource

L'objet DataSource définit une source de données générique.

## Explication

Cet objet permet de référencer des objets de la collection DataSources. Comme l'objet DataSource représente une source de données générique, les propriétés changent de façon à refléter la source de données sous-jacente.

L'objet DataSource ne possède pas de constante de type dans la liste des valeurs de xtrObjectType. Il dérive la valeur de la propriété Type de la source de données qu'il représente. En raison de cette particularité ainsi que d'autres différences, cet objet n'est pas aussi efficace que des objets spécifiques de la source de données, tels que les objets IqdDataSource.

Pour exploiter toutes les fonctionnalités disponibles avec l'objet DataSource, utilisez une instruction similaire à celle ci-dessous dans Visual Basic ou tout autre éditeur compatible VB :

```
Dim objDataSource As CognosTransformer.DataSource
```

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet DataSource.
Méthode Move	Déplace l'objet DataSource vers une autre position dans la collection DataSources.
Méthode Update	Met à jour l'objet DataSource.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Description	Définit ou retourne la description de l'objet DataSource.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.

Propriété	Description
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom unique de la source de données.
Propriété Parent	Retourne l'objet parent.
Propriété ServerQuery	Définit ou retourne une valeur indiquant si les données sont traitées localement ou sur un serveur.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété SourceType	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété Type	Retourne le type de la source de données sous-jacente que référence actuellement l'objet DataSource.

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
```

---

## Objet DateDimension

L'objet DateDimension organise les données de date dans des groupes logiques.

### Explication

Une dimension de temps est une structure hiérarchique qui représente des informations de date relatives à un aspect important d'une entreprise. Chaque dimension de temps est constituée d'une collection DrillDowns et d'un ou plusieurs objets DateDrillDown, qui représentent des branches de niveaux.

Utilisez cet objet pour spécifier le contexte de date de n'importe quel modèle. Si vous n'incluez pas au moins un objet DateDimension, vous perdez la plupart des avantages offerts par un cube, tels que le suivi des bénéfices et des coûts. En outre, en l'absence d'une dimension de temps, la conversion monétaire est désactivée.

Quand vous utilisez la méthode DoAutoDesign, Transformer crée automatiquement une dimension de temps en fonction des définitions de date contenues dans la source de données.

Vous pouvez créer manuellement un objet DateDimension de deux manières :

- Définissez toutes les propriétés applicables de l'objet DateWizard et affectez une source de données à l'aide de la méthode DimensionAssociateWith. Utilisez ensuite la méthode CreateDateDimension pour terminer le processus. Celui-ci

créé la dimension ainsi qu'une branche de niveaux et des niveaux de date. Il génère aussi un ensemble de catégories de dates associées, telles que l'année à ce jour.

- Utilisez la méthode Add de la collection Dimensions, puis affectez la valeur trDateDimension au paramètre Type. Vous devez définir la propriété Name de l'objet. Ensuite, vous pouvez affecter une source de données à l'aide de la méthode AssociateWith. Cette procédure crée la branche de niveaux à laquelle vous pouvez ensuite ajouter des niveaux de date et des catégories de dates associées.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AssociateWith	Associe un objet source à un objet DateDimension.
Méthode CleanHouse	Supprime les catégories inactives d'une dimension de temps.
Méthode Delete	Supprime l'objet DateDimension.
Méthode GenerateDateCategories	Remplit un modèle avec les catégories de date spécifiées par StartDate et EndDate.
Méthode Move	Déplace la dimension de temps à une autre position dans la collection Dimensions.
Méthode SetAllocation	Modifie le type d'allocation de la mesure utilisée par l'objet.
Méthode Update	Met à jour l'objet DateDimension.

Propriété	Description
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Associations	Retourne une collection Associations.
Propriété CalculationDefinitions	Retourne une collection d'objets CalculationDefinition.

Propriété	Description
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété Categories	Retourne la collection des catégories qui sont incluses dans l'objet DateDimension.
Propriété CategoryCount	Retourne le nombre de catégories ordinaires de l'objet DateDimension.
Propriété Description	Définit ou retourne la description de l'objet DateDimension.
Propriété DimensionLevels	Retourne une collection DimensionLevels.
Propriété DrillDowns	Retourne une collection DrillDowns.
Propriété EarliestDate	Définit ou retourne la date la plus ancienne dans une plage de catégories de dates.
Propriété ExcludeAutoPartition	Définit ou retourne une valeur indiquant si la dimension de temps est exclue du processus de découpage automatique en partitions.
Propriété GenerateTimePeriod	Définit ou retourne les options de génération de catégorie pour une dimension de temps.
Propriété GetDefaultCategory	Retourne la catégorie par défaut appartenant à une dimension.
Propriété LatestDate	Définit ou retourne la date la plus récente dans une plage de catégories de dates.
Propriété ManualCurrentPeriod	Définit ou retourne une valeur indiquant si la période de date en cours est définie manuellement ou par Transformer.
Propriété Name	Définit ou retourne le nom de l'objet DateDimension.

Propriété	Description
Propriété NewCatsLocked	Définit ou retourne une valeur indiquant si vous pouvez ajouter de nouvelles catégories à l'objet DateDimension.
Propriété Parent	Retourne l'objet parent.
Propriété SetDefaultCategory	Spécifie une nouvelle catégorie par défaut pour une dimension, différente de la catégorie racine ou parent d'un groupe de catégories de scénarios.
Propriété SpecialCategoryCount	Retourne le nombre de changements de niveaux, de racines et de catégories spéciales dans une dimension.
Propriété Type	Retourne le type de l'objet.
Propriété Views	Retourne une collection Views.
Propriété WorkingDay	Définit ou retourne une valeur indiquant si un jour fait partie de la semaine de travail.
Propriété WorkingDays	Définit ou retourne une valeur indiquant les jours qui font partie de la semaine de travail.

## Exemples

```
objDateDim = objModel.Dimensions("Date")
```

---

## Objet DateDrillDown

L'objet DateDrillDown définit des branches de niveaux principales et secondaires pour des valeurs de date ou de temps.

### Explication

Cet objet permet de modifier les propriétés de l'objet DateDrillDown principal ou de définir des objets DateDrillDown secondaires.

Chaque dimension de temps possède une collection DrillDowns. Quand vous créez une dimension de temps, Transformer ajoute automatiquement un objet DateDrillDown principal à cette collection. Une collection ne contient qu'une seule branche principale de niveaux que vous ne pouvez pas supprimer. Tous les autres objets DateDrillDown de la collection sont des objets DateDrillDown secondaires que vous créez. Une branche de niveaux secondaire définit une autre perspective des données dans PowerPlay.

Vous pouvez convertir une branche de niveaux secondaire en branche principale à l'aide de la propriété IsPrimary.

Pour ajouter un objet DateDrillDown à la collection, retournez l'objet DateLevel à utiliser comme niveau de détail le plus bas dans la nouvelle branche de niveaux.

Utilisez ensuite la méthode `CreateAlternateDrillDown` de l'objet `DateLevel` spécifié pour créer la nouvelle branche de niveaux. Comme ce niveau est commun à la branche existante et aux nouvelles branches de niveaux, il porte le nom de niveau de convergence.

Pour ajouter d'autres niveaux de date à la branche de niveaux secondaire, utilisez la méthode `Add` du nouvel objet `DateDrillDown` que vous venez de créer. Vous devez ensuite modifier l'ordre de la collection pour refléter la hiérarchie de niveaux requise. Pour cela, utilisez la méthode `Move` afin de déplacer les niveaux supplémentaires vers une position située au-dessus du niveau de convergence. Si le niveau de convergence correspond à `Weeks`, par exemple, et si vous ajoutez le niveau `Years` à la collection `Levels`, utilisez la méthode `Move` pour déplacer l'objet du niveau `Years` à la première position de la collection.

Les branches de niveaux secondaires débutent toujours à la catégorie racine et se développent vers le niveau de convergence sélectionné. Pour connecter correctement plusieurs catégories parents à la même catégorie de convergence, chaque catégorie d'un niveau de convergence doit être unique et non ambiguë. Pour spécifier la caractéristique unique de la catégorie, affectez la valeur `True` à la propriété `Unique` du niveau de convergence.

Comme les valeurs des catégories au niveau de convergence et aux niveaux inférieurs sont partagées par toutes les branches de niveaux, la suppression ou la modification d'une catégorie dans une branche au niveau de convergence et aux niveaux inférieurs affecte immédiatement la même catégorie dans toutes les autres branches de niveaux.

Les catégories qui appartiennent à plusieurs branches de niveaux possèdent plusieurs catégories parents. Utilisez la propriété `ParentCategories` pour retourner une collection de catégories parents.

Pour en savoir davantage sur les branches de niveaux secondaires dans les dimensions de temps, reportez-vous à l'aide en ligne de `Transformer`.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode `Update`.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode <code>Delete</code>	Supprime l'objet <code>DateDrillDown</code> .
Méthode <code>Update</code>	Met à jour l'objet <code>DateDrillDown</code> .

Propriété	Description
Propriété <code>Application</code>	Retourne l'objet <code>Application Transformer</code> .
Propriété <code>Categories</code>	Retourne la collection des catégories qui sont incluses dans l'objet <code>DateDrillDown</code> .

Propriété	Description
Propriété ConvergenceLevel	Retourne le niveau de convergence de la branche de niveaux secondaire.
Propriété DrillCode	Définit ou retourne un code qui identifie de manière unique la catégorie de la branche de niveaux dans toute la dimension.
Propriété DrillInclusion	Définit ou retourne une valeur indiquant si l'objet DateDrillDown est inclus dans le cube ou non.
Propriété IsPrimary	Définit ou retourne une valeur indiquant si la branche de niveaux est la branche principale.
Propriété Levels	Retourne une collection Levels.
Propriété Lunar	Définit ou retourne une valeur indiquant si l'objet est basé sur une année lunaire.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété ShortName	Définit ou retourne le nom abrégé de la branche de niveaux.
Propriété Type	Retourne le type de l'objet.
Propriété WeekAdd	Définit ou retourne le nombre de jours qui sont ajoutés à une année lunaire.
Propriété WeekSpan	Définit ou retourne une valeur indiquant la manière dont il faut traiter une semaine qui s'étale sur deux années.
Propriété WeekStart	Définit ou retourne le premier jour de la semaine.
Propriété YearStartDay	Définit ou retourne le premier jour d'une année.

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales
regions")objLevel = objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDown
```

---

## Objet DateLevel

L'objet DateLevel définit un niveau de date dans un modèle.



## Explication

L'objet DateLevel permet de présenter le contexte de date des données du PowerCube, notamment les intervalles annuels, trimestriels ou mensuels.

Chaque objet DateDimension d'un modèle inclut une collection Levels qui contient des objets DateLevel. Les deux types de niveaux sont les suivants :

- Les niveaux sources contiennent des catégories qui sont générées à partir de la source de données. Chaque niveau source est lié à une colonne de la source de données par l'intermédiaire d'un objet Association.
- Les niveaux manuels ne sont associés à aucune colonne de la source. Vous pouvez, par exemple, créer un niveau manuel destiné à contenir des catégories isolées.

Utilisez la méthode Add de la collection Levels pour créer un objet DateLevel.

Utilisez la méthode CreateAlternateDrillDown pour convertir un niveau de date en niveau de convergence dans une branche de niveaux secondaire.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Collection	Description
Collection LevelCategories	Retourne une collection de catégories présentes dans un niveau.

Méthode	Description
Méthode AssociateWith	Associe un objet source à un objet DateLevel.
Méthode CreateAlternateDrillDown	Crée une branche de niveaux secondaire dans une dimension et convertit le niveau en niveau de convergence.
Méthode Delete	Supprime l'objet DateLevel.
Méthode Move	Déplace l'objet DateLevel à une autre position dans la collection Levels.
Méthode SetAllocation	Modifie le type d'allocation de la mesure utilisée par l'objet.
Méthode Update	Met à jour l'objet DateLevel.

<b>Propriété</b>	<b>Description</b>
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Associations	Retourne une collection Associations.
Propriété BlankSubstitute	Définit ou retourne le libellé par défaut des catégories en blanc générées dans le niveau de date.
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux de date descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux de date descendants et aux catégories.
Propriété CategoryCount	Retourne le nombre de catégories ordinaires de l'objet DateLevel.
Propriété DateFormat	Définit ou retourne une valeur indiquant la manière dont les dates apparaissent.
Propriété DateFunction	Définit ou retourne les catégories de date qui sont générées dans un niveau.
Propriété Description	Définit ou retourne la description de l'objet DateLevel.
Propriété GenerateDateCategories	Définit ou retourne une valeur indiquant si un niveau de date génère des catégories de date.
Propriété HasSubdimension	Retourne une valeur indiquant si un objet de niveau contient une sous-dimension.
Propriété Inclusion	Définit ou retourne une valeur indiquant à quel moment une catégorie est incluse dans un cube.

<b>Propriété</b>	<b>Description</b>
Propriété IsManual	Retourne une valeur indiquant si le niveau de date a été créé manuellement ou généré.
Propriété LevelDrillDowns	Retourne la collection LevelDrillDowns.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété NewCatsLocked	Définit ou retourne une valeur indiquant si vous pouvez ajouter de nouvelles catégories à l'objet DateLevel.
Propriété OrderByDescending	Définit ou retourne une valeur indiquant si les valeurs apparaissent dans l'ordre croissant.
Propriété OrderByStorageType	Définit ou retourne une valeur indiquant la manière dont les catégories sont triées en fonction du type de stockage d'une colonne.
Propriété Parent	Retourne l'objet parent.
Propriété Partition	Définit ou retourne le numéro d'une partition manuelle.
Propriété QualifiedName	Retourne le nom complet du niveau de date.
Propriété RefreshDescription	Définit ou retourne une valeur indiquant si les descriptions sont mises à jour.
Propriété RefreshLabel	Définit ou retourne une valeur indiquant si les libellés sont mis à jour.
Propriété RefreshShortName	Définit ou retourne une valeur indiquant si les noms abrégés sont mis à jour.
Propriété TimeRank	Définit ou retourne le rang relatif des niveaux de date dans une dimension de temps.
Propriété Type	Retourne le type de l'objet.
Propriété Unique	Définit ou retourne une valeur indiquant si Transformer peut identifier chaque catégorie du niveau de date à l'aide d'une valeur source unique.
Propriété UniqueMove	Définit ou retourne une valeur indiquant la manière dont un objet DateLevel est traité en cas de déplacement de la catégorie associée.

## Exemples

```
objDateDim = objModel.Dimensions("Date")objDateLevel  
= objDateDim.DimensionLevels("Month")
```

---

## Objet DateWizard

L'objet DateWizard crée un objet DateDimension.

### Explication

Une dimension de temps est une structure hiérarchique qui représente des informations de date relatives à une entreprise. Cet objet permet de créer aisément des objets DateDimension et DateLevel.

Vous devez d'abord utiliser la propriété DateWizard de l'objet Model pour retourner l'objet DateWizard. Définissez ensuite toutes les propriétés applicables de l'objet DateWizard et affectez une source de données à l'aide de la méthode DimensionAssociateWith. Enfin, utilisez la méthode CreateDateDimension pour créer un objet CreateDateDimension. Pour activer une hiérarchie lunaire, définissez les propriétés MonthType, QuarterType et YearType sur des valeurs lunaires.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode CreateDateDimension	Crée un objet DateDimension basé sur les propriétés de l'objet DateWizard.
Méthode DimensionAssociateWith	Associe un objet source au nouvel objet DateDimension.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété DimensionName	Définit le nom du nouvel objet DateDimension.
Propriété EarliestDate	Définit la date la plus ancienne dans une plage de catégories de dates.
Propriété EnableTimePeriod	Définit le niveau de détail d'une dimension de temps.
Propriété GeneratePowerCube	Définit une valeur indiquant si l'objet DateWizard génère des catégories de date.
Propriété LatestDate	Définit la date la plus ancienne dans une plage de catégories de dates.

Propriété	Description
Propriété MonthType	Définit le mode de calcul du niveau mois d'une dimension de temps.
Propriété Parent	Retourne l'objet parent.
Propriété QuarterType	Définit le mode de calcul du niveau trimestre d'une dimension de temps.
Propriété Type	Retourne le type de l'objet.
Propriété WeekAdd	Définit le nombre de jours qui sont ajoutés à une année lunaire.
Propriété WeekSpan	Détermine la manière dont il faut traiter une semaine qui s'étale sur deux années.
Propriété WeekStartDay	Définit le premier jour de la semaine.
Propriété WorkingDay	Définit une valeur indiquant si un jour fait partie de la semaine de travail.
Propriété WorkingDays	Définit les jours qui font partie de la semaine de travail.
Propriété YearStartDay	Définit le premier jour d'une année.
Propriété YearType	Définit le mode de calcul du niveau année d'une dimension de temps.

## Exemples

```
objDateWizard = objModel.DateWizard
```

---

## Objet DbDataSource

L'objet DbDataSource représente un fichier externe au format de base de données.

### Explication

Cet objet permet la connexion à un fichier externe au format de base de données.

Pour créer un objet DbDataSource, utilisez la méthode Add de la collection DataSources. Assurez-vous que vous affectez la constante trDbDataSource au paramètre Type. Affectez ensuite à la propriété SourceType le type de fichier de base de données à ouvrir. Spécifiez l'emplacement du fichier de données à l'aide des propriétés LocalPath et ServerPath applicables.

Les objets DbDataSource contiennent une collection Columns. Les colonnes correspondent à la source des niveaux dans une dimension.

Transformer peut utiliser des informations de base de données à partir de :

- dBase

- Excel
- Accès
- Paradox
- FoxPro
- Clipper

Vous pouvez, par exemple, utiliser cette propriété pour spécifier une table à partir d'une base de données Access (.mdb).

Vous pouvez utiliser une requête Access qui désigne une source de données ODBC pour récupérer des données du serveur dans Transformer. Si vous utilisez une requête Access comme source, la collection Columns est dérivée de la requête SQL qui référence la base de données.

Si vous utilisez des tables Paradox avec une clé de tri autre que la clé ASCII, qui est le paramètre par défaut du moteur Jet pour Paradox, vous devez inclure le fichier d'index primaire (.px) au même emplacement que le fichier de base de données Paradox (.db) avant d'importer les tables. Vous devez aussi remplacer la clé du registre Windows par une valeur qui correspond à la clé de tri utilisée lors de la création de la table. Les valeurs correctes sont les suivantes :

- ASCII,
- International
- Norvégien-Danois,
- Japonais
- Suédois-Finnois.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet DbDataSource.
Méthode Move	Déplace l'objet DbDataSource à une autre position dans la collection DataSources.
Méthode Update	Met à jour l'objet DbDataSource.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Columns	Retourne une collection Columns.
Propriété ColumnsLoaded	Retourne une valeur indiquant si les colonnes d'une source de données ont été utilisées pour créer un modèle.

Propriété	Description
Propriété DataRange	Définit ou retourne le nom d'une plage de base de données dans la source de données.
Propriété DecimalPoint	Définit ou retourne le caractère de séparation utilisé comme virgule décimale.
Propriété Description	Définit ou retourne la description de l'objet DbDataSource.
Propriété External	Définit ou retourne une valeur indiquant si les valeurs de la source de données sont récapitulées ou cumulées lorsqu'elles sont utilisées par le modèle.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété LocalPath	Définit ou retourne l'emplacement de la source de données locale.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom unique de la source de données.
Propriété Parent	Retourne l'objet parent.
Propriété ServerPath	Définit ou retourne le nom et l'emplacement d'un fichier de la source de données.
Propriété ServerQuery	Renvoie une valeur indiquant si les données sont traitées localement ou sur un serveur.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété SourceType	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété ThousandPoint	Définit ou retourne le caractère qui sépare les milliers.

Propriété	Description
Propriété Type	Retourne le type de l'objet. Le type d'objet renvoyé est trDbQuery même si le type xtrObjectType.trDbDataSource doit être utilisé pour ajouter et extraire un élément depuis la collection DataSources.

## Objet Dimension

L'objet Dimension organise les données autres que de type date dans des groupes logiques.

### Explication

L'objet Dimension permet de regrouper des objets Level et DrillDown qui fournissent des données numériques ou des données de texte. Une dimension est une structure hiérarchique qui représente des informations relatives à un aspect important d'une entreprise.

Si vous utilisez la méthode DoAutoDesign, Transformer crée automatiquement des dimensions et leurs objets Level et DrillDown associés en fonction des modèles et des relations dans les sources de données.

Vous pouvez aussi créer manuellement des objets Dimension. Pour cela, utilisez d'abord la méthode Add de la collection Dimensions. Ajoutez ensuite des objets de niveau aux collections applicables de la dimension.

Pour les données de date ou de temps, utilisez plutôt un objet DateDimension.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AssociateWith	Associe un objet source à un objet Dimension.
Méthode CleanHouse	Supprime les catégories inactives d'une dimension.
Méthode Delete	Supprime l'objet Dimension.
Méthode FindCategoryByCatCode	Retourne l'objet Category contenant la chaîne du code de catégorie spécifiée.
Méthode GetDefaultCategory	Retourne la catégorie par défaut appartenant à une dimension.
Méthode Move	Déplace la dimension à une autre position dans la collection Dimensions.
Méthode SetAllocation	Modifie le type d'allocation de la mesure utilisée par l'objet.



Méthode	Description
Méthode SetDefaultCategory	Spécifie une nouvelle catégorie par défaut pour une dimension, différente de la catégorie racine ou parent d'un groupe de catégories de scénarios.
Méthode Update	Met à jour l'objet Dimension.

Propriété	Description
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Associations	Retourne une collection Associations.
Propriété CalculationDefinitions	Retourne une collection d'objets CalculationDefinition.
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété Categories	Retourne la collection des catégories spéciales qui sont incluses dans l'objet Dimension.
Propriété CategoryCount	Retourne le nombre de catégories ordinaires de l'objet Dimension.
Propriété Description	Définit ou retourne la description de l'objet Dimension.
Propriété DimensionLevels	Retourne une collection DimensionLevels.
Propriété DrillDowns	Retourne une collection DrillDowns.
Propriété ExcludeAutoPartition	Définit ou retourne une valeur indiquant si la dimension est exclue du processus de découpage automatique en partitions.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété NewCatsLocked	Définit ou retourne une valeur indiquant si vous pouvez ajouter de nouvelles catégories à l'objet Dimension.
Propriété Parent	Retourne l'objet parent.
Propriété SpecialCategoryCount	Retourne le nombre de changements de niveaux, de racines et de catégories spéciales dans une dimension.
Propriété Type	Retourne le type de l'objet.
Propriété Views	Retourne une collection Views.

## Exemples

```
objDimensions = objModel.DimensionsobjLocationsDim  
= objDimensions.Item(3)
```

---

## Objet DrillDown

L'objet DrillDown définit des branches de niveaux principales et secondaires pour naviguer dans les cubes.

### Explication

Une collection DrillDowns peut contenir une branche principale de niveaux. Toutes les autres branches de niveaux de la collection DrillDowns sont des branches secondaires que vous créez. Une branche de niveaux secondaire définit une autre perspective des données dans PowerPlay.

Cet objet permet de modifier les propriétés de la branche principale de niveaux ou de définir des branches de niveaux secondaires. Vous ne pouvez pas supprimer la branche principale de niveaux créée par Transformer.

Pour ajouter un nouvel objet DrillDown secondaire à la collection DrillDowns, appliquez la méthode CreateAlternateDrillDown à l'objet de niveau que vous souhaitez utiliser comme niveau de convergence dans la branche de niveaux secondaire. Le niveau que vous sélectionnez devient le premier niveau de la collection Levels du nouvel objet DrillDown. Vous ajoutez ensuite d'autres niveaux à cette collection Levels à l'aide de la méthode Add. Comme la méthode Add ajoute toujours les nouveaux niveaux à la fin de la collection, utilisez la méthode Move pour les déplacer vers une position qui précède le niveau de convergence.

Les branches de niveaux secondaires débutent toujours à la catégorie racine et se développent vers le niveau de convergence sélectionné. Pour connecter correctement plusieurs catégories parents à la même catégorie de convergence, chaque catégorie d'un niveau de convergence doit être unique et non ambiguë. Cela signifie que deux catégories du niveau ne peuvent jamais avoir la même valeur. Pour que Transformer accepte une branche de niveaux secondaire, vous devez affecter la valeur True à la propriété Unique du niveau de convergence.

Comme les valeurs des catégories au niveau de convergence et aux niveaux inférieurs sont partagées par toutes les branches de niveaux, la suppression ou la modification d'une catégorie dans une branche au niveau de convergence et aux niveaux inférieurs affecte immédiatement la même catégorie dans toutes les autres branches de niveaux.

Les catégories qui appartiennent à plusieurs branches de niveaux possèdent plusieurs catégories parents. Utilisez la propriété ParentCategories pour retourner une collection de catégories parents.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet DrillDown.
Méthode Update	Met à jour l'objet DrillDown.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Categories	Retourne la collection des catégories qui sont incluses dans l'objet DrillDown.
Propriété ConvergenceLevel	Retourne le niveau de convergence de la branche de niveaux secondaire.
Propriété DrillCode	Définit ou retourne un code qui identifie de manière unique la catégorie de la branche de niveaux dans toute la dimension.
Propriété DrillInclusion	Définit ou retourne une valeur indiquant si l'objet DrillDown est inclus dans le cube ou non.
Propriété IsPrimary	Définit ou retourne une valeur indiquant si la branche de niveaux est la branche principale.
Propriété Levels	Retourne une collection Levels.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété ShortName	Définit ou retourne le nom abrégé de la branche de niveaux.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objDrill = objModel.Dimensions("Retailers").DrillDowns(2)
```

---

## Objet DrillThroughTarget

L'objet DrillThroughTarget représente une liaison d'accès au détail d'un rapport externe ou d'un PowerCube.

## Explication

Les cibles d'accès au détail peuvent représenter des rapports Impromptu (.imr), des fichiers de définition de requête d'Impromptu (.iqd), des rapports PowerPlay (.ppr), des fichiers PowerCubes (.mdc) ou des fichiers de script de macro (.mac). Grâce à l'automatisation OLE, l'application associée s'ouvre directement à partir de PowerPlay et affiche le rapport depuis l'emplacement spécifié.

Utilisez la méthode Add de la collection DrillThroughTargets pour créer un objet DrillThroughTarget. Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Measure, vous limitez la fonction d'accès au détail à cette seule mesure. Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Cube ou CubeGroup, la fonction d'accès au détail est disponible à tout endroit du cube ou des cubes.

Vous devez affecter la valeur True à la propriété AllowDrillThrough avant de pouvoir accéder à la fonction d'accès au détail.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet DrillThroughTarget.
Méthode isExcludeDimension	Retourne une valeur indiquant si la Dimension donnée est exclue.
Méthode isExcludeLevel	Retourne une valeur indiquant si le Niveau donné est exclu.
Méthode SetExcludeDimension	Définit la dimension donnée à la valeur isRestricted.
Méthode SetExcludeLevel	Définit le Niveau donné à la valeur isRestricted.
Méthode Update	Met à jour l'objet DrillThroughTarget.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Description	Définit ou retourne la description de l'objet DrillThroughTarget.
Propriété Name	Définit ou retourne le nom de l'objet. Pour un rapport, il doit s'agir d'un nom de fichier complet.
Propriété Parent	Retourne l'objet parent.

Propriété	Description
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objReport = objMeasure.DrillThroughTargets.Add(strReportPath,
"Default Report")
```

---

## Objet Filter

L'objet Filter représente un filtre dans le modèle Transformer.

### Explication

Une collection Filters est renvoyée par la propriété Filters (objet Query). Chaque objet Filter de la collection fait référence à un filtre dans le modèle Transformer.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Filter.
Méthode Update	Met à jour l'objet Filter.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété RefName	Définit ou retourne le nom du filtre dans la source de données.

## Exemples

```
new_package = model.Packages.Add()new_query
= new_package.Queries.Add()new_filter = new_query.Filters.Add()
```

---

## Objet FlatFileDataSource

L'objet FlatFileDataSource représente un fichier externe au format ASCII.

### Explication

Utilisez un objet FlatFileDataSource quand vos données sont au format ASCII et que les colonnes sont délimitées par des séparateurs de la zone.

Pour créer un objet FlatFileDataSource, utilisez la méthode Add de la collection DataSources. Affectez la constante trFlatFileDataSource au paramètre Type. Affectez ensuite à la propriété SourceType le type de fichier à plat à ouvrir. Décrivez le format et la structure des données à l'aide des propriétés telles que FieldSeparator, DecimalPoint et ThousandPoint. Spécifiez l'emplacement du fichier de données à l'aide de la propriété LocalPath ou ServerPath.

Les objets FlatFileDataSource contiennent une collection Columns. Les colonnes correspondent à la source des niveaux dans une dimension.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet FlatFileDataSource.
Méthode Move	Déplace l'objet FlatFileDataSource à une autre position dans la collection DataSources.
Méthode Update	Met à jour l'objet FlatFileDataSource.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété CharacterType	Définit ou retourne le type du jeu de caractères utilisé par la source de données.
Propriété Columns	Retourne une collection Columns.
Propriété ColumnsLoaded	Retourne une valeur indiquant si les colonnes d'une source de données ont été utilisées pour créer un modèle.
Propriété DecimalPoint	Définit ou retourne le caractère utilisé comme virgule décimale dans le fichier à plat.
Propriété Description	Définit ou retourne la description de l'objet FlatFileDataSource.
Propriété External	Définit ou retourne une valeur indiquant si la source de données contient des valeurs récapitulatives.
Propriété FieldSeparator	Retourne le caractère utilisé pour marquer les limites des zones dans le fichier à plat.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.

Propriété	Description
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété LocalPath	Définit ou retourne l'emplacement de la source de données locale.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom unique de la source de données.
Propriété Parent	Retourne l'objet parent.
Propriété ServerPath	Définit ou retourne le nom et l'emplacement d'un fichier de la source de données.
Propriété ServerQuery	Définit ou retourne une valeur indiquant si les données sont traitées localement ou sur un serveur.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété SourceType	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété ThousandPoint	Définit ou retourne le caractère qui sépare les milliers.
Propriété Type	Retourne le type de l'objet. Un type <code>trFlatFileQuery</code> est renvoyé, cependant, un <code>trFlatFileDataSource</code> doit être utilisé lors de l'extraction ou de l'ajout d'une source de données au niveau de la collection <code>DataSources</code> .

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
```

---

## Objet IqdDataSource

L'objet `IqdDataSource` représente un fichier de définition de requête d'Impromptu (.iqd).

### Explication

Cet objet permet de définir l'accès à un fichier externe au format Impromptu.

Pour créer un objet IqdDataSource, utilisez la méthode Add de la collection DataSources. Assurez-vous que vous affectez la constante trIqdDataSource au paramètre Type.

Affectez la constante trQuery à la propriété SourceType de l'objet IqdDataSource. Spécifiez l'emplacement du fichier de définition de requête d'Impromptu à l'aide des propriétés LocalPath ou ServerPath applicables.

Les objets IqdDataSource contiennent une collection Columns. Les colonnes correspondent à la source des niveaux dans une dimension.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet IqdDataSource.
Méthode Move	Déplace l'objet IqdDataSource à une autre position dans la collection DataSources.
Méthode Update	Met à jour l'objet IqdDataSource.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Columns	Retourne une collection Columns.
Propriété ColumnsLoaded	Retourne une valeur indiquant si les colonnes d'une source de données ont été utilisées pour créer un modèle.
Propriété DecimalPoint	Définit ou retourne le caractère utilisé comme virgule décimale dans le fichier Iqd.
Propriété Description	Définit ou retourne la description de l'objet IqdDataSource.
Propriété External	Définit ou retourne une valeur indiquant si les valeurs de la source de données sont récapitulées ou cumulées lorsqu'elles sont utilisées par le modèle.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.



Propriété	Description
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété IsolationLevel	Définit ou retourne le niveau d'isolement utilisé pour définir des transactions autorisées.
Propriété LocalPath	Définit ou retourne l'emplacement de la source de données locale.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom unique de la source de données.
Propriété Parent	Retourne l'objet parent.
Propriété ServerPath	Définit ou retourne le nom et l'emplacement d'un fichier de la source de données.
Propriété ServerQuery	Définit ou retourne une valeur indiquant si les données sont traitées localement ou sur un serveur.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété SourceType	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété SQLExpression	Retourne l'expression SQL utilisée pour définir un fichier de définition de requête d'Impromptu (.iqd).
Propriété ThousandPoint	Définit ou retourne le caractère qui sépare les milliers.
Propriété Type	Retourne le type de l'objet. trIqdQuery est le type renvoyé même si le type trIqdDataSource doit être utilisé lors de l'extraction ou de l'ajout au niveau de la collection DataSources.

## Exemples

```
datasource = model.DataSources.Add(xtrObjectType.trIqdDataSource)
```

---

## Objet de niveau

L'objet de niveau définit un niveau dans une dimension.

### Explication

L'objet de niveau permet de présenter toutes les données de PowerCube autres que celles qui sont directement associées à des dates.

Chaque objet Dimension d'un modèle inclut une collection Levels qui contient des objets de niveau. Les deux types de niveaux sont les suivants :

- Les niveaux sources contiennent des catégories qui sont générées à partir de la source de données. Chaque niveau source est lié à une colonne de la source de données par l'intermédiaire d'un objet Association.
- Les niveaux manuels ne sont associés à aucune colonne de la source. Vous pouvez, par exemple, créer un niveau manuel destiné à contenir des catégories isolées.

Utilisez la méthode Add de la collection Levels pour créer un objet de niveau.

Utilisez la méthode CreateAlternateDrillDown pour convertir un niveau en niveau de convergence dans une branche de niveaux secondaire.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AssociateWith	Associe un objet source à l'objet de niveau.
Méthode CreateAlternateDrillDown	Crée une branche de niveaux secondaire dans une dimension.
Méthode Delete	Supprime l'objet de niveau.
Méthode Move	Déplace le niveau à une autre position dans la collection Levels.
Méthode SetAllocation	Modifie le type d'allocation de la mesure utilisée par l'objet.
Méthode Update	Met à jour l'objet de niveau.

Propriété	Description
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.

Propriété	Description
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Associations	Retourne une collection Associations.
Propriété BlankSubstitute	Définit ou retourne le libellé par défaut des catégories en blanc générées dans l'objet de niveau.
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété CategoryCount	Retourne le nombre de catégories ordinaires de l'objet de niveau.
Propriété Description	Définit ou retourne la description de l'objet de niveau.
Propriété HasSubdimension	Retourne une valeur indiquant si un objet de niveau contient une sous-dimension.
Propriété Inclusion	Définit ou retourne une valeur indiquant à quel moment une catégorie est incluse dans un cube.
Propriété IsManual	Retourne une valeur indiquant si un niveau est associé à une valeur source.
Propriété LevelCategories	Retourne une collection de catégories présentes dans un niveau.
Propriété LevelDrillDowns	Retourne une collection LevelDrillDowns.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété NewCatsLocked	Définit ou retourne une valeur indiquant si vous pouvez ajouter de nouvelles catégories à l'objet de niveau.

Propriété	Description
Propriété OrderByDescending	Définit ou retourne une valeur indiquant si les valeurs apparaissent dans l'ordre croissant.
Propriété OrderByStorageType	Définit ou retourne une valeur indiquant la manière dont les catégories sont triées en fonction du type de stockage d'une colonne.
Propriété Parent	Retourne l'objet parent.
Propriété Partition	Définit ou retourne le numéro d'une partition manuelle.
Propriété QualifiedName	Définit ou retourne le nom complet de l'objet de niveau.
Propriété RefreshDescription	Définit ou retourne une valeur indiquant si les descriptions sont mises à jour.
Propriété RefreshLabel	Définit ou retourne une valeur indiquant si les libellés sont mis à jour.
Propriété RefreshShortName	Définit ou retourne une valeur indiquant si les noms abrégés sont mis à jour.
Propriété Type	Retourne le type de l'objet.
Propriété Unique	Définit ou retourne une valeur indiquant si Transformer peut identifier chaque catégorie du niveau à l'aide d'une valeur source unique.
Propriété UniqueMove	Définit ou retourne une valeur indiquant la manière dont un objet de niveau unique est traité en cas de déplacement de la catégorie associée.

## Exemples

```
objTimeDimension = objModel.Dimensions.Item("Time").objLevel
= objTimeDimension.DrillDowns.Item(1).Levels.Item("Month")
```

---

## Objet Measure

L'objet Measure représente des valeurs quantitatives dans un cube.

### Explication

Une mesure permet d'identifier des ensembles de valeurs numériques dans un modèle. Les mesures sont quantifiables et fournissent des indicateurs de performances dans PowerPlay.

Vous pouvez créer trois types d'objets de mesure : ordinaire, calculé et nombre de catégories. Chaque type de mesure génère des valeurs d'une manière différente. Une mesure ordinaire utilise un objet Association. Une mesure calculée utilise une expression définie dans la propriété ExpressionText. Une mesure de nombre de catégories utilise le paramètre des propriétés CategoryCountLevel et ActivityMeasure.

Une mesure créée par Transformer est toujours une mesure ordinaire. Pour créer une mesure calculée ou une mesure de nombre de catégories, utilisez la méthode Add de la collection Measures. Utilisez ensuite les propriétés CategoryCountLevel et ActivityMeasure pour spécifier une mesure de nombre de catégories ou bien la propriété ExpressionText pour définir une mesure calculée.

Aux niveaux inférieurs, les valeurs de mesure sont cumulées et aux niveaux supérieurs, elles sont récapitulées. Utilisez les propriétés de cumul de cet objet pour spécifier la manière dont les valeurs de mesure sont cumulées. Les valeurs peuvent notamment faire l'objet d'une addition, d'une moyenne ou d'une pondération.

Dans Transformer Series 7 Version4 et les éditions ultérieures, vous pouvez définir que les valeurs nulles et manquantes doivent être ignorées lorsque les mesures d'état temporel Moyenne ou Moyenne pondérée sont cumulées. Si les valeurs manquantes sont représentées par la mention "ND", définissez la propriété IgnoreMissingValue de l'objet objMeasure sur TRUE lorsque vous créez ou mettez à jour la définition d'un type de mesure pris en charge.

**Remarque :** vous devez conserver la valeur par défaut pour la première période, la dernière période et la période en cours. Les valeurs nulles et manquantes ne peuvent être exclues des calculs de cumul pour ces types de mesures. Les valeurs de données manquantes (null) sont toujours exclues des calculs de cumul Min et Max, qu'elle soient définies par Transformer pour afficher "0" ou "n/d" (paramètre ND).

En présence de plusieurs sources de données, vous pouvez utiliser la méthode SetAllocation pour définir la manière dont les valeurs de mesure associées à une source de données sont affichées dans les catégories associées à une autre source de données.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AddToFolder	Ajoute une mesure à un dossier de mesures.
Méthode AssociateWith	Associe un objet source à un objet Measure.
Méthode Delete	Supprime l'objet Measure.
Méthode Move	Déplace la mesure à une autre position dans la collection Measures.

<b>Méthode</b>	<b>Description</b>
Méthode RemoveFromFolder	Supprime une mesure enfant du dossier des mesures actif.
Méthode Update	Met à jour l'objet Measure.

<b>Propriété</b>	<b>Description</b>
Propriété ActivityMeasure	Définit ou retourne la mesure utilisée par une mesure de nombre de catégories.
Propriété AllowCurrencyConversion	Définit ou retourne une valeur indiquant si vous pouvez modifier une devise.
Propriété AllowDrillThrough	Définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Associations	Retourne une collection Associations.
Propriété CategoryCountLevel	Définit ou retourne l'objet de niveau auquel s'applique une mesure de nombre de catégories.
Propriété ChildMeasures	envoie une collection de mesures enfant qui proviennent du dossier de mesures. Renvoie une collection vide si un dossier de mesures ne contient aucune mesure enfant. Une exception est générée si la mesure n'est pas un dossier de mesures.
Propriété Description	Définit ou retourne la description de l'objet Measure.
Propriété DrillThroughTargets	Retourne une collection d'objets cible d'accès au détail associés à un modèle Transformer.
Propriété DuplicateRollup	Définit ou retourne une valeur indiquant la manière dont les valeurs de mesure des enregistrements consolidés sont cumulées.
Propriété DuplicateWeight	Définit ou retourne le nom de la mesure qui contient des facteurs de pondération de moyenne.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur d'une mesure calculée. Dans l'édition en cours, les expressions conditionnelles if-then-else sont désormais prises en charge.

<b>Propriété</b>	<b>Description</b>
Propriété Format	Définit ou retourne une valeur indiquant la manière dont les valeurs numériques apparaissent.
Propriété FormatDecimals	Définit ou retourne le nombre de décimales que PowerPlay affiche pour la mesure.
Propriété IgnoreMissingValue	Spécifie si les valeurs nulles et manquantes doivent être ignorées dans un cumul d'état temporel. Valeur par défaut : false.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété IsFolder	Définit et renvoie une valeur qui indique si une mesure représente un dossier de mesures.
Propriété Label	Définit ou retourne un nom descriptif qui apparaît dans PowerPlay.
Propriété MeasureType	Retourne une valeur indiquant s'il s'agit d'une mesure ordinaire, d'une mesure calculée ou d'une mesure de nombre de catégories.
Propriété MissingValue	Définit ou retourne une valeur indiquant ce qui est affiché à la place d'une valeur vide ou nulle.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété OutputScale	Définit ou retourne la valeur d'échelle utilisée pour convertir les nombres entiers en nombres décimaux dans PowerPlay.
Propriété Parent	Retourne l'objet parent.
Propriété Precision	Définit ou retourne le nombre de décimales des mesures utilisées dans des calculs.
Propriété RegularRollup	Définit ou retourne la fonction de cumul ordinaire en cours de l'objet Measure.
Propriété RegularWeight	Définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.
Propriété ReverseSign	Définit ou retourne une valeur indiquant si PowerPlay inverse le signe d'une mesure.

Propriété	Description
Propriété RollupTiming	Définit ou retourne une valeur indiquant à quel moment les calculs des mesures calculées doivent être effectués.
Propriété ShortName	Définit ou retourne le nom abrégé de l'objet Measure.
Propriété StorageType	Définit ou retourne la taille d'un type de données numérique.
Propriété TimeStateRollup	Définit ou retourne la période de date utilisée pour les cumuls d'état temporel.
Propriété TimeStateWeight	Définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objMeasure = objModel.Measures("Forecast")
```

---

## Objet Model

L'objet Model définit un modèle Transformer.

### Explication

Le modèle est l'objet de plus haut niveau dans un objet Application. A partir de cet objet, vous pouvez ouvrir directement toutes les collections de niveau supérieur.

Pour créer un modèle et un PowerCube, suivez les étapes ci-après :

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Collection	Description
Collection Namespaces	Contient tous les objets Namespace d'un modèle.
Collection SecurityObjects	Contient tous les objets SecurityObject d'un modèle.

Méthode	Description
Méthode CheckLocalPowerCubes	Contrôle les cubes définis dans le modèle par rapport aux fichiers de PowerCube (.mdc) associés.



<b>Méthode</b>	<b>Description</b>
Méthode CheckModel	Exécute une vérification manuelle du modèle.
Méthode CleanHouse	Supprime les catégories inactives du modèle.
Méthode Close	Ferme le modèle.
Méthode CreateMDCFiles	Crée des fichiers de PowerCube (.mdc) en fonction de toutes les définitions de cubes et de groupes de cubes du modèle.
Méthode DeleteAllCustomViews	Supprime toutes les vues personnalisées relatives au modèle.
Méthode DeleteAllSecurityObjects	Supprime les objets de sécurité du modèle.
Méthode DeployCubes	Déploie tous les PowerCubes modèles vers tous les emplacements de déploiement.
Méthode DoAutoDesign	Génère des dimensions, des niveaux, des branches de niveaux, des cubes et des mesures.
Méthode GenerateCategories	Remplit un modèle avec des catégories.
Méthode LoadCurrencyTable	Charge une table des devises dans le modèle.
Méthode ResetPartitions	Supprime les partitions du cube en cours.
Méthode Save	Enregistre le modèle.
Méthode SaveAs	Enregistre le modèle sous un nouveau nom.
Méthode TestBuild	Crée un petit modèle de test ou un cube.
Méthode Update	Met à jour l'objet Model.

<b>Propriété</b>	<b>Description</b>
Propriété CubeCodePage	Définit ou retourne le paramètre de page de codes du cube du modèle utilisé pour créer le cube.
Propriété Application	Retourne l'objet Application Transformer.
Propriété Cubes	Retourne une collection d'objets Cube et CubeGroup.
Propriété CurrencyRecords	Retourne une collection CurrencyRecords.

Propriété	Description
Propriété CurrencyTables	Retourne une collection CurrencyTables.
Propriété DataSources	Retourne une collection DataSources.
Propriété DateWizard	Retourne l'objet DateWizard.
Propriété DefaultCategoryOrderBy	Définit l'ordre de tri par défaut (croissant) pour toutes les catégories du modèle.
Propriété Description	Définit ou retourne la description de l'objet Model.
Propriété Dimensions	Retourne une collection Dimensions.
Propriété FileName	Renvoie le nom d'un fichier modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété FullName	Retourne le nom et l'emplacement d'un fichier modèle.
Propriété Measures	Retourne une collection Measures.
Propriété ModelType	Renvoie l'extension d'un fichier modèle tel qu'elle apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Namespaces	Renvoie une collection Namespaces.
Propriété Packages	Renvoie une collection Packages.
Propriété Parent	Retourne l'objet parent.
Propriété Path	Retourne l'emplacement d'un fichier modèle.
Propriété SecurityObjects	Renvoie la collection SecurityObjects.
Propriété ServerModelPath	Définit ou retourne l'emplacement du serveur d'un modèle.
Propriété Signon	Retourne une collection Signon.
Propriété Size	Spécifie la taille du modèle en octets.
Propriété Time	Renvoie l'horodateur d'un modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objModel = objTransApp.OpenModel(strModelPath)
```

---

## Objet Name

L'objet Name fait référence à une chaîne de texte représentant un nom ou un message.

### Explication

Une collection Names est renvoyée par la méthode CheckModel (objet Model).

Chaque objet Name de cette collection contient une chaîne de message complète. Vous pouvez utiliser la valeur de retour de la propriété Name pour lire les résultats de la procédure CheckModel.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Name	Retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
MsgBox(objModel.CheckModel(intX).Name)
```

---

## Objet Namespace

L'objet Namespace représente un espace-noms dans le modèle Transformer.

### Explication

Une collection Namespace est renvoyée par la propriété Namespaces dans un objet Model.

Chaque objet Namespace d'une collection fait référence à un espace-noms dans le modèle Transformer.

Les espaces-noms sont utilisés pour créer des objets Security. Les objets Security sont ajoutés à CustomViews pour fournir une sécurité au niveau utilisateur et groupe dans un PowerCube Transformer.

Pour pouvoir utiliser un objet Namespace, vous devez d'abord vous connecter à l'espace-noms applicable, à l'aide de la méthode Logon de l'objet Application. Il doit fournir un espace-noms, un nom d'utilisateur et un mot de passe.

Pour utiliser un objet Namespace, la propriété Name doit être définie sur l'espace-noms ou la propriété ID doit être définie sur l'ID de l'espace-noms. Si la sécurité est appliquée à un utilisateur, la propriété User doit aussi être définie sur le nom de l'utilisateur. Si la sécurité est appliquée à un autre objet, comme un

groupe ou un rôle, la propriété ObjectName doit être définie sur le nom de l'objet. Après avoir défini ces propriétés de façon appropriée, d'autres propriétés peuvent être utilisées pour extraire des informations de sécurité.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Namespace.
Méthode Update	Met à jour l'objet Namespace.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété CAMID	Renvoie le CAMID de l'espace-noms.
Propriété ID	Définit ou retourne l'ID d'espace-noms.
Propriété Name	Renvoie le nom de l'espace-noms.
Propriété ObjectCAMID	Renvoie le CAMID de l'objet dans l'ensemble d'espaces-noms par la propriété ObjectName.
Propriété ObjectName	Définit ou retourne le nom d'un objet NameSpace.
Propriété Parent	Retourne l'objet parent.
Propriété SecurityObjects	Renvoie une collection de SecurityObjects.
Propriété User	Définit ou retourne un nom d'utilisateur associé à l'espace- noms.
Propriété UserCAMID	Renvoie le CAMID de l'utilisateur défini par la propriété User.

## Exemples

```
new_namespace = model.Namespaces.Add()
```

---

## Objet Package

L'objet Package représente une source de données de pack dans le modèle Transformer.

## Explication

Une collection Package est renvoyée par la propriété Packages de l'objet Model. Chaque objet Package dans la collection référence une source de données de pack dans le modèle Transformer.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Package.
Méthode Update	Met à jour l'objet Package.
Méthode Verify	Vérifie l'objet Package.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété PackagesDatasourceConnections	Renvoie la collection d'objets PackageDatasourceConnection associés au pack.
Propriété Parent	Retourne l'objet parent.
Propriété Path	Définit ou renvoie le chemin d'accès au pack dans IBM Cognos Connection.
Propriété Queries	Renvoie la collection d'objets Query associés au pack.
Propriété TimeStamp	Définit ou renvoie la date et l'heure du pack.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
new_package = model.Packages.Add()
```

---

## Objet PackageDatasourceConnection

L'objet PackageDatasourceConnection représente une connexion de source de données de pack dans le modèle Transformer.

## Explication

Lorsqu'un modèle est basé sur une source de données qui compte plusieurs connexions ou plusieurs codes d'accès, il existe une ambiguïté qui doit être résolue.

Le système recherche dans le modèle une connexion de source de données de pack susceptible de résoudre l'ambiguïté. Dans ce cas, ce sont la source de données, la connexion et le code d'accès de source de données qui sont utilisés.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet PackageDataSourceConnection.
Méthode Update	Met à jour l'objet PackageDataSourceConnection.
Méthode Verify	Vérifie l'objet PackageDataSourceConnection.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété AlwaysUseTransformerSignon	Lorsque la valeur est définie à Vrai, le système utilise le code d'accès de Transformer avant celui de Content Manager. Lorsque la valeur est définie à Faux, le code d'accès de Content Manager est utilisé par défaut.
Propriété Connection	Définit ou retourne la connexion à Content Manager.
Propriété DataSource	Définit ou retourne la source de données de Content Manager.
Propriété Parent	Retourne l'objet parent.
Propriété Signon	Définit ou retourne le code d'accès à Content Manager.
Propriété TransformerSignon	Définit ou renvoie l'objet de code d'accès Transformer associé à une connexion de source de données de pack, qui correspond au code d'accès IBM Cognos.

## Exemples

```
package = model.Packages.Add() connection
= package.PackageDataSourceConnections.Add()
```

---

## Objet Prompt

L'objet Prompt représente une invite dans le modèle Transformer.

## Explication

Une collection Prompts est renvoyée par la propriété Prompts (objet Query). Chaque objet Prompt dans la collection référence une invite dans le modèle Transformer.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Query.
Méthode Update	Met à jour l'objet Query.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété CurrentValueIndex	Définit ou retourne l'index de valeur d'invite en cours. Cette propriété est utilisée pour effectuer une itération dans les valeurs d'invite.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété PromptValueType	Définit ou retourne le type d'invite.
Propriété Type	Définit ou retourne le type de données de la ou des valeurs d'invite.
Propriété Value	Définit ou retourne la valeur d'invite.
Propriété ValuesCount	Renvoie le nombre d'ensembles de valeurs pour l'invite.

## Exemples

```
new_report = model.Reports.Add()new_query = new_report.Queries.Add()new_prompt  
= new_query.Prompts.Add()
```

---

## Objet Query

L'objet Query représente une requête faisant partie d'un rapport ou pack.

## Explication

Une collection Query est renvoyée par la propriété Queries de l'objet Package ou Report. Chaque objet Package et Report contient des objets de requête. Ces objets de requête contiennent des objets de colonne qui référencent des colonnes dans le pack sous-jacent ou la source de données de rapport. Pour enregistrer les

modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Query.
Méthode Update	Met à jour l'objet Query.
Méthode Verify	Vérifie l'objet Query.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété AutoSummary	Définit ou retourne l'option AutoSummary pour une requête.
Propriété Columns	Retourne une collection Columns qui contient tous les objets Column d'une source de données.
Propriété Filters	Retourne un objet Filter.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété Prompts	Renvoie la collection d'objets Prompt associés à une requête.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.



Propriété	Description
Propriété StreamExtractAllowed	Définit ou retourne une valeur indiquant si l'extraction de flux est autorisée. (SAP BW uniquement).
Propriété StreamExtractSize	Définit ou renvoie la taille, en mégaoctets, du tampon utilisé pour transférer des données depuis SAP BW quand StreamExtract est défini sur True.
Propriété SuppressNull	Définit ou renvoie l'option de suppression des valeurs nulles utilisée pour les sources de données SAP BW.

## Exemples

```
new_package = model.Packages.Add()new_query
= new_package.Queries.Add()
```

---

## Objet Report

L'objet Report représente une source de données de rapport dans le modèle Transformer.

### Explication

Une collection Reports est renvoyée par la propriété Reports (objet Model). Chaque objet Report dans la collection référence une source de données de rapport dans le modèle Transformer.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet Report.
Méthode Update	Met à jour l'objet Report.
Méthode Verify	Vérifie l'objet Query.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété PackagesDatasourceConnections	Renvoie la collection d'objets PackageDatasourceConnection associés au rapport.

Propriété	Description
Propriété Parent	Retourne l'objet parent.
Propriété Path	Renvoie le chemin d'accès au rapport dans IBM Cognos Connection.
Propriété Queries	Renvoie la collection d'objets Query associés au rapport.
Propriété TimeStamp	Définit ou renvoie l'horodatage du rapport.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
new_report = model.Reports.Add()
```

---

## Objet SecurityObject

L'objet SecurityObject représente un objet de sécurité dans le modèle Transformer.

### Explication

Une collection SecurityObjects est renvoyée par la propriété SecurityObjects dans un objet Namespace. Chaque objet SecurityObject dans la collection fait référence à un objet de sécurité dans le modèle Transformer.

Les espaces-noms sont utilisés pour créer des objets Security. Les objets Security sont ajoutés à CustomViews pour fournir une sécurité au niveau utilisateur et groupe dans un PowerCube Transformer. La propriété Name de SecurityObject est définie sur un CAMID d'un objet User, Group ou Role. La propriété Type doit être définie sur une constante xtrSecurityType.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode AddToCustomView	Ajoute un objet de sécurité à une vue personnalisée.
Méthode Delete	Supprime l'objet SecurityObject.
Méthode Update	Met à jour l'objet SecurityObject.
Méthode Verify	Vérifie l'objet SecurityObject.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.

Propriété	Description
Propriété DisplayName	Définit ou retourne le nom à afficher pour l'objet SecurityObject.
Propriété Name	Définit ou retourne le nom (CAMID) de l'objet SecurityObject.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
new_namespace = model.Namespaces.Add()
securityObject = new_namespace.SecurityObjects.Add()
```

---

## Objet de code d'accès

L'objet de code d'accès définit le type de code d'accès, qui peut être une source de données ou Cognos. Le code d'accès de type source de données, précédemment appelé code d'accès à la base de données dans IBM Cognos Series 7, s'applique à un code d'accès à la base de données. Le code d'accès IBM Cognos est sécurisé. Il est associé à un espace-noms sécurisé et est exclusivement utilisé dans IBM Cognos.

Il n'existe qu'un seul objet de code d'accès. La propriété SignonType d'un objet de code d'accès indique si vous utilisez un type DataSourceSignon ou CognosSignon.

## Explication

- Code d'accès à la source de données  
Transformer utilise les propriétés Name, Password et User ID d'un objet de code d'accès pour se connecter automatiquement à une base de données.
- Fichiers Impromptu  
Les fichiers de définition de requête d'Impromptu (.iqd) référencent des sources de base de données. Ces fichiers utilisent un nom de base de données logique pour représenter les paramètres de connexion à une source de données sous-jacente. Lorsque vous ajoutez un fichier de définition de requête d'Impromptu à votre modèle, Transformer ajoute automatiquement un objet de code d'accès qui contient le nom de base de données logique. Il peut aussi contenir les informations relatives au mot de passe et à l'ID utilisateur.  
Une fois que vous avez retourné un objet de code d'accès à partir de la collection Signons, utilisez-le pour éditer le mot de passe et l'ID utilisateur.  
Les objets de code d'accès ajoutés par Transformer ne peuvent pas être supprimés de la collection.
- Code d'accès IBM Cognos  
Transformer utilise le type CognosSignon pour l'authentification auprès d'un espace-noms sécurisé.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet de code d'accès.
Méthode Update	Met à jour l'objet de code d'accès.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété AutoLogon	Indique si l'application s'authentifie automatiquement auprès des espaces-noms sécurisés associés au code d'accès. S'applique uniquement au type CognosSignon.
Propriété Description	Définit ou retourne une description pour l'objet.
Propriété Name	Retourne le nom de l'objet de code d'accès.
Propriété Parent	Retourne l'objet parent.
Propriété Password	Définit un mot de passe pour l'objet de code d'accès (prenant en compte la distinction majuscule/minuscule).
Propriété PromptForPassword	Indique si l'utilisateur est toujours invité à saisir un mot de passe lorsqu'il utilise l'interface utilisateur de Transformer. S'applique uniquement au type DatasourceSignon.
Propriété SignOnNamespace	Contient l'espace-noms sécurisé associé au type CognosSignon.
Propriété SignonType	Définit le type de code d'accès (en l'occurrence, il s'agit de xtrSignonType).
Propriété Type	Retourne le type de l'objet.
Propriété UserID	Définit ou retourne l'ID utilisateur de l'objet de code d'accès.

## Exemples

```
signon = model.Signons.Add()
```

---

## Objet SpecialCategory

L'objet SpecialCategory définit une catégorie avec une valeur qui ne provient pas directement d'une source de données.

### Explication

Cet objet permet de créer une catégorie en fonction d'une date relative, telle que l'année à ce jour.

Une collection d'objets SpecialCategory est retournée par un objet Dimension. Pour créer un objet SpecialCategory, utilisez la méthode Add de la collection Categories et affectez au paramètre Type la valeur trSpecialCategory.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

### Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode ConnectWithCategory	Déplace une catégorie enfant vers une nouvelle catégorie parent dans le même niveau.
Méthode Delete	Supprime l'objet SpecialCategory.
Méthode SetAllocation	Modifie le type d'allocation de la mesure utilisée par l'objet.
Méthode Update	Met à jour l'objet SpecialCategory.

Propriété	Description
Propriété Aggregate	Définit ou retourne le type de catégorie de date relative.
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété Application	Retourne l'objet Application Transformer.
Propriété AutoLogon	Indique si l'application s'authentifie automatiquement auprès des espaces-noms sécurisés associés au code d'accès. S'applique uniquement au type CognosSignon.

Propriété	Description
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété ChildCategories	Retourne une collection Categories.
Propriété Code	Définit ou retourne le code unique d'une catégorie spéciale au sein d'une dimension.
Propriété ContextLevel	Définit ou retourne une période de date.
Propriété ContextOffset	Définit ou retourne la position de la période de référence par rapport à la période en cours.
Propriété Description	Définit ou retourne la description de l'objet SpecialCategory.
Propriété Dimensions	Retourne la dimension d'un objet SpecialCategory.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur de l'objet SpecialCategory.
Propriété Format	Définit ou retourne une valeur indiquant la manière dont les valeurs numériques apparaissent.
Propriété Inclusion	Définit ou retourne une valeur indiquant à quel moment une catégorie est incluse dans un cube.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété Label	Définit ou retourne le nom de l'objet. Dans l'interface utilisateur de Transformer, le libellé est "Valeur source", mais la valeur par défaut est "".

Propriété	Description
Propriété LastUseDate	Retourne la date de la dernière modification ou de la dernière utilisation de l'objet SpecialCategory.
Propriété Level	Retourne le niveau d'un objet SpecialCategory.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété ParentCategories	Retourne une collection ParentCategories.
Propriété ReverseSign	Définit ou retourne une valeur indiquant si PowerPlay inverse le signe d'une mesure.
Propriété Rollup	Définit ou retourne une valeur indiquant si les valeurs de mesure de la catégorie spéciale sont cumulées dans la catégorie parent.
Propriété RunningPeriods	Définit ou retourne le nombre de périodes de date utilisées dans le total cumulatif.
Propriété ShortName	Définit ou retourne le nom abrégé de l'objet SpecialCategory.
Propriété TargetLevel	Définit ou retourne le niveau de détail d'une période de date.
Propriété TargetOffset	Définit ou retourne la position de la période cible par rapport à la période en cours.
Propriété ToDateLevel	Définit ou retourne la période de date utilisée pour les totaux jusqu'à ce jour.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objSpecCategory =
_ objModel.Dimensions("Time").Categories.Add(xtrObjectType.trSpecialCategory)
```

---

## Objet SuspendedModel

L'objet SuspendedModel représente un modèle incomplet existant.

### Explication

Cet objet permet de récupérer un modèle qui ne se ferme pas correctement (notamment dans le cas d'un incident système ou d'une panne secteur).

Transformer recherche les modèles en attente à l'emplacement spécifié par la propriété ModelTemporaryFilePath ou ModelsPath. Si les deux propriétés sont définies, Transformer ne vérifie que l'emplacement spécifié par ModelTemporaryFilePath. Si aucune propriété n'est définie, il vérifie l'emplacement temporaire spécifié par la variable d'environnement.

Utilisez la propriété IsBad pour vérifier si le modèle en attente est altéré ou récupérable. S'il est altéré, utilisez la méthode RemoveSuspendedModel pour le supprimer de la collection SuspendedModels. S'il est récupérable, vous pouvez l'ouvrir dans Transformer et déterminer son statut.

Transformer conserve le plus d'informations possible dans un objet SuspendedModel. Transformer ajoute des entrées de point de contrôle après chaque étape principale du processus de création de modèle. Ces points de contrôle sont consignés dans un fichier temporaire dont l'extension commence par "qy?". Etant donné que ces fichiers temporaires sont supprimés lorsque le modèle se ferme normalement, la présence d'un fichier modèle temporaire indique que Transformer s'est arrêté de façon inattendue. (L'extension de fichier porte toujours un troisième caractère qui varie.)

Transformer consigne aussi des messages dans un fichier journal (.log) qui est stocké au même endroit que vos modèles ou à un emplacement spécifié par l'utilisateur. Si Transformer n'a pas la possibilité de reprendre automatiquement après une erreur de traitement ou si vous choisissez d'ignorer le traitement précédent et de recommencer le processus, vous pouvez consulter le fichier journal pour déterminer la cause de l'erreur.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Propriété	Description
Propriété Application	Retourne l'objet Application Transformer.
Propriété IsBad	Retourne une valeur indiquant si le modèle en attente est corrompu.
Propriété ModelName	Retourne le nom du modèle en attente.
Propriété Parent	Retourne l'objet parent.
Propriété QyPath	Retourne le chemin d'accès d'un modèle en attente.
Propriété Type	Retourne le type de l'objet.

## Exemples

```
objSuspendedModel = objTransApp.SuspendedModels(intX)
```

---

## Objet View

L'objet View définit une vue partielle d'une dimension.

## Explication

L'objet View permet de présenter aux utilisateurs de PowerPlay un sous-ensemble des informations d'une dimension.



Il existe des vues de dimensions et des vues personnalisées. Les deux vues comprennent tout, partie ou aucun des niveaux ou catégories contenus dans une dimension.

- Vue des dimensions

Chaque objet Dimension inclut automatiquement une collection de deux objets View. Par défaut, la propriété DimensionView a pour valeur le premier objet de la collection, à savoir: "Toutes les catégories". Pour omettre une dimension à partir d'un cube, affectez à la propriété DimensionView le deuxième objet de la collection, à savoir : Omettre la dimension.

Pour créer une vue personnalisée d'une dimension, utilisez d'abord la méthode Add pour ajouter un objet View à la collection, puis affectez la valeur trViewTypeCustom à la propriété ViewType. Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété Apex ou associer l'objet View aux catégories ou aux niveaux sélectionnés au moyen de la méthode SetViewStatus. Une fois que vous avez défini l'objet View, associez-le au cube à l'aide de la propriété DimensionView.

- Vue personnalisée

Quand vous incluez la sécurité dans un modèle, chaque objet CustomView inclut automatiquement une collection d'objets View, à savoir un pour chaque dimension. La vue par défaut de chaque objet d'une collection est la vue "Toutes catégories". Vous pouvez modifier cette vue à l'aide de la propriété DimensionInclude en omettant toutes les catégories ou en créant une vue personnalisée.

Si vous spécifiez une vue personnalisée à l'aide de la propriété DimensionInclude, vous devez utiliser ensuite la propriété DimensionView pour retourner l'objet View associé à une dimension. Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété Apex ou associer l'objet View aux catégories ou aux niveaux sélectionnés au moyen de la méthode SetViewStatus. Pour terminer une vue personnalisée, vous devez associer une vue personnalisée à un cube. Pour cela, utilisez la méthode Add afin d'ajouter un objet CustomView à la collection CubeCustomViews du cube.

Pour enregistrer les modifications apportées aux propriétés de cet objet, utilisez la méthode Update.

## Rubriques connexes

Ces tables répertorient les collections, méthodes et propriétés connexes.

Méthode	Description
Méthode Delete	Supprime l'objet View.
Méthode SetViewStatus	Définit de quelle façon un objet Category, SpecialCategory, de niveau ou DateLevel est affiché.
Méthode Update	Met à jour l'objet View.

Propriété	Description
Propriété Apex	Définit ou retourne l'objet Category utilisé comme racine de la vue d'une dimension.
Propriété Application	Retourne l'objet Application Transformer.
Propriété CustomViews	Définit ou retourne la vue personnalisée associée à une vue.
Propriété Name	Définit ou retourne le nom de l'objet.
Propriété Parent	Retourne l'objet parent.
Propriété Type	Retourne le type de l'objet.
Propriété ViewType	Définit ou retourne une valeur indiquant si une vue contient la totalité des catégories d'une dimension, une partie seulement ou aucune d'entre elles.

### Exemples

```
objDimension = objModel.Dimensions("Retailers")objView
= objDimension.Views.Add()
```

---

## Chapitre 5. Méthodes

Le tableau suivant récapitule tous les objets d'automatisation OLE de Transformer :

Méthode	Description
Méthode Add ()	Ajoute un objet à une collection.
Méthode Add (Categories)	Ajoute un objet Category ou SpecialCategory à une collection Categories.
Méthode Add (CustomViews)	Ajoute une vue personnalisée à la liste des utilisateurs qui peuvent utiliser un cube.
Méthode Add (DrillThroughTargets)	Ajoute un nouveau rapport personnalisé d'accès au détail de la collection Reports d'un objet Cube, CubeGroup ou Measure.
Méthode Add (Objects)	Ajoute un objet à une collection qui contient des objets de plusieurs types.
Méthode AddDeployLocation	Ajoute un emplacement de déploiement pour la copie et l'activation.
Méthode AddToCustomView	Ajoute un objet de sécurité à une vue personnalisée.
Méthode AddToFolder	Ajoute une mesure à un dossier de mesures.
Méthode AssociateWith	Associe un objet source à un objet du modèle et ajoute un objet Association à la collection Associations.
Méthode CheckLocalPowerCubes	Contrôle les cubes définis dans le modèle par rapport aux fichiers de PowerCube (.mdc) associés.
Méthode CheckModel	Vérifie si le modèle en cours contient des objets en conflit avec la source de données, ou les uns avec les autres.
Méthode CleanHouse	Supprime les catégories inactives d'un modèle ou d'une dimension.
Méthode ClearDeployLocations	Supprime tous les emplacements de déploiement.
Méthode Close	Ferme le modèle actif.

Méthode	Description
Méthode ConnectWithCategory	Déplace une catégorie enfant vers une nouvelle catégorie parent dans le même niveau.
Méthode CreateAlternateDrillDown	Crée une branche de niveaux secondaire dans une dimension et convertit le niveau en niveau de convergence.
Méthode CreateDateDimension	Crée un objet DateDimension basé sur les propriétés de l'objet DateWizard.
Méthode CreateMDCFile	Crée des fichiers PowerCube (.mdc) pour un cube unique ou pour tous les cubes d'un groupe de cubes.
Méthode CreateMDCFiles	Crée tous les fichiers PowerCube (.mdc) dans un modèle.
Méthode Delete	Supprime un objet.
Méthode DeleteAllCustomViews	Supprime toutes les vues personnalisées relatives à un modèle.
Méthode DeleteAllSecurityObjects	Supprime les objets de sécurité d'un modèle.
Méthode DeployCube	Déploie un PowerCube vers tous les emplacements de déploiement.
Méthode DeployCubes	Déploie tous les PowerCubes d'un modèle vers tous les emplacements de déploiement.
Méthode DimensionAssociateWith	Associe un objet source au nouvel objet DateDimension et ajoute un objet Association à la collection Associations.
Méthode DoAutoDesign	Génère des dimensions, des niveaux, des branches de niveaux, des cubes et des mesures.
Méthode FindCategoryByCatCode	Retourne l'objet Category contenant la chaîne du code de catégorie spécifiée.
Méthode GenerateCategories	Remplit un modèle avec des catégories.
Méthode GenerateDateCategories	Remplit un modèle avec des catégories de date.
Méthode GetDefaultCategory	Retourne la catégorie par défaut appartenant à une dimension.
Méthode GetViewStatus	Retourne le statut d'affichage d'un objet Category, SpecialCategory, de niveau ou DateLevel.

<b>Méthode</b>	<b>Description</b>
Méthode IsExcludeDateDimension	Retourne une valeur indiquant si une dimension de date particulière est exclue.
Méthode IsExcludeDateLevel	Retourne une valeur indiquant si le niveau de date donné est exclu.
Méthode IsExcludeDimension	Retourne une valeur indiquant si une dimension est limitée ou non.
Méthode IsExcludeLevel	Retourne une valeur indiquant si un niveau est limité ou non.
Méthode Item	Retourne un objet spécifique dans une collection.
Méthode Item ()	Retourne un objet spécifique dans une collection.
Méthode Item ()	Retourne un objet spécifique dans une collection.
Méthode LoadCurrencyTable	Charge une table des devises dans le modèle.
Méthode Logoff	Se déconnecte de tous les espaces-noms.
Méthode Logon	Se connecte à un espace-noms.
Méthode Move	Réorganise des objets dans une collection.
Méthode MoveToCategory	Déplace une catégorie enfant vers une autre catégorie parent.
Méthode MoveToLevel	Déplace une catégorie enfant vers un nouveau niveau de la catégorie parent en cours.
Méthode NewModel	Crée un modèle.
Méthode OpenModel	Ouvre un modèle existant.
Méthode OpenSuspendedModel	Ouvre un modèle en attente.
Méthode PublishDatasource	Publie la connexion à la source de données pour un PowerCube.
Méthode PublishPackage	Publie la source de données et le pack pour un PowerCube.
Méthode RemoveCubeCustomView	Supprime la vue personnalisée du cube ou du cube enfant.

<b>Méthode</b>	<b>Description</b>
Méthode Remove	Supprime un objet spécifique d'une collection.
Méthode RemoveFromFolder	Supprime une mesure enfant du dossier des mesures actif.
Méthode RemoveSuspendedModel	Supprime un modèle en attente.
Méthode ResetPartitions	Supprime les partitions du cube en cours.
Méthode Save	Enregistre les modifications du modèle en cours.
Méthode SaveAs	Enregistre le modèle en cours sous un autre nom de fichier.
Méthode SetAllocation	Modifie le type d'allocation d'une mesure utilisée par un objet.
Méthode SetDefaultCategory	Spécifie une nouvelle catégorie par défaut pour une dimension (autre que la catégorie racine).
Méthode SetDeployType	Définit le type de déploiement pour la copie et l'activation.
Méthode SetExcludeDateDimension	Définit la dimension de date donnée à la valeur isRestricted.
Méthode SetExcludeDateLevel	Définit le niveau de date donnée à la valeur isRestricted.
Méthode SetExcludeDimension	Pour un objet Report et Dimension donné, définit la limite d'accès au détail à True si la valeur du paramètre isRestricted est True.
Méthode SetExcludeLevel	Pour un objet Report et Level donné, définit la limite d'accès au détail à True si la valeur du paramètre isRestricted est True.
Méthode SetViewStatus	Retourne le statut d'affichage d'un objet Category, SpecialCategory, de niveau ou DateLevel.
Méthode TestBuild	Crée un petit modèle de test ou un cube.
Méthode Update	Met à jour l'objet associé.
Méthode Verify	Vérifie l'objet associé.

---

## Méthode Add ()

La méthode Add ajoute un objet à une collection.

### Syntaxe

*collection* .Add()

### Application

Collection Associations

Collection CalculationDefinitions

Collections CategorySets

Collection Columns

Collection CurrencyRecords

Collection CurrencyTables

Collection CustomViews

Collection Filters

Collection PackageDatasourceConnections

Collection Packages

Collection Queries

Collection Measures

Collection Namespaces

Collection Reports

Collection SecurityObjects

Collection Signons

Collection Views

### Explication

Utilisez cette méthode sans paramètre pour ajouter des objets à leurs collections respectives.

Vous pouvez aussi utiliser les méthodes AssociateWith et DimensionAssociateWith pour ajouter des objets à une collection Associations.

### Type de retour

Objet

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()
```

---

## Méthode Add (Categories)

La méthode Add ajoute un objet Category ou SpecialCategory à une collection Categories.

### Syntaxe

```
Categories .Add([Type] [, Level ])
```

### Application

Collection Categories

### Explication

Quand vous ajoutez un objet à la collection Categories d'un objet CategorySet, le paramètre Type doit nommer un objet Category ou SpecialCategory qui existe déjà. Pour toutes les autres collections Categories, le paramètre Type fournit une valeur de xtrObjectType ou une Variant qui fournit une valeur de xtrObjectType.

Utilisez le paramètre facultatif Niveau pour spécifier le niveau auquel appartient la catégorie.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Type	Facultatif. Pour la plupart des collections, spécifie la constante trCategory ou trSpecialCategory de xtrObjectType. Dans le cas de la collection Categories d'un objet CategorySet, ce paramètre nomme un objet existant.  Type : Variante
Niveau	Facultatif. Spécifie le nom du niveau auquel la catégorie doit être ajoutée. En cas d'omission, la catégorie est ajoutée au niveau le plus bas.  Type : Objet

### Type de retour

Objet

### Exemples

```
objSpecCategory =  
_ objModel.Dimensions("Time").Categories.Add(xtrObjectType.trSpecialCategory)
```



---

## Méthode Add (CustomViews)

La méthode Add ajoute une vue personnalisée à la liste des utilisateurs qui peuvent utiliser un cube.

### Syntaxe

```
CubeCustomViews .Add(CustomView)
```

### Application

CubeCustomViews

### Explication

Avant d'utiliser cette propriété, créez un objet CustomView, puis définissez et mettez à jour ses propriétés.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
CustomView	Requise. Définit une vue personnalisée dans le modèle Transformer.  Type : vue personnalisée

### Type de retour

CustomView

### Exemples

```
cube = model.Cubes.Item("Sales and Marketing") cube.CubeCustomViews.Add(custom_view)
```

---

## Méthode Add (DrillThroughTargets)

La méthode Add (DrillThroughTargets) ajoute un nouveau DrillThroughTarget personnalisé d'accès au détail à la collection DrillThroughTargets d'un objet Cube, CubeGroup, ChildCube ou Measure.

### Syntaxe

```
DrillThroughTargets .Add(DrillThroughTargetName, Description)
```

### Application

Collection DrillThroughTargets

### Discussion

Les objets DrillThroughTarget peuvent représenter des rapports Impromptu (.imr), des fichiers de définition de requête d'Impromptu (.iqd), des rapports PowerPlay (.ppr), des fichiers PowerCubes (.mdc) ou des fichiers de script de macro (.mac).

Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Measure, vous limitez la fonctionnalité d'accès au détail à cette seule mesure. Quand vous ajoutez un objet DrillThroughTarget à la collection DrillThroughTargets d'un objet Cube, ChildCube ou CubeGroup, la fonctionnalité d'accès au détail est disponible en n'importe quel point d'un rapport PowerPlay. Le paramètre DrillThroughTargetName définit la propriété DrillThroughTarget.Name. Un nom de fichier complet doit être donné lors de la définition de la propriété DrillThroughTarget.Name. Cette règle s'applique également au paramètre Description. Elle est définie en fonction de la valeur transmise en tant que paramètre Description.

Paramètre	Description
DrillThroughTargetName	Requise. Spécifie un nom de fichier complet pour la cible d'accès au détail.  Type : chaîne
Description	Requise. Spécifie une description textuelle succincte du but de la cible d'accès au détail à l'utilisateur de PowerPlay.  Type : chaîne

## Type de retour

Objet - DrillThroughTarget

## Exemples

```
objReport = objMeasure.DrillThroughTargets.Add(strReportPath,
"Default Report")
```

---

## Méthode Add (Objects)

La méthode Add (Objects) ajoute un objet à une collection qui contient des objets de plusieurs types.

### Syntaxe

```
collection .Add(Type)
```

### Application

Collection Cubes

Collection DataSources

Collection Dimensions

Collection Levels

## Explication

Utilisez cette méthode lorsqu'une collection contient des objets de plusieurs types, comme dans la collection `DataSources`. Le paramètre `Type` spécifie une valeur de la liste des valeurs de `xtrObjectType`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Type	<p>Facultatif pour les objets <code>DataSource</code>.</p> <p><b>Remarque</b> : Si le type <code>DataSource</code> n'est pas spécifié, le type par défaut est un objet <code>FlatFileDataSource</code>.</p> <p>Facultatif pour les autres objets. Spécifie une constante de la liste des valeurs de <code>xtrObjectType</code>. Requis lors de l'ajout de <code>DateDimensions</code> et quand un type de <code>trDateDimension</code> doit être utilisé.</p> <p>Type : Variante</p>

## Type de retour

Objet

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
```

---

## Méthode AddDeployLocation

La méthode `AddDeployLocation` ajoute un emplacement de déploiement pour la copie et l'activation.

### Syntaxe

```
Cube.AddDeployLocation(deployPath)
```

### Application

Objet `Cube`

### Explication

Cette méthode permet de configurer le déploiement d'un `PowerCube`. Elle peut être appelée plusieurs fois pour ajouter plusieurs emplacements de déploiement à l'objet `Cube`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
deployPath	Spécifie un chemin vers le répertoire dans lequel le PowerCube doit être déployé. Type : chaîne

## Type de retour

Aucun

## Exemples

```
cube = model.Cubes.Item("National")cube.AddDeployLocation("c:\NATIONAL\Deployment1")
```

---

## Méthode AddToCustomView

La méthode AddToCustomView ajoute un objet de sécurité à une vue personnalisée.

## Syntaxe

*SecurityObject* .AddToCustomView

## Application

Objet SecurityObject

## Explication

Utiliser lors de l'ajout d'un objet SecurityObject à un objet CustomView. Utiliser pour ajouter un utilisateur, un groupe ou un objet de sécurité à une vue personnalisée.

Le paramètre CustomView spécifie l'objet CustomView auquel est ajouté l'objet de sécurité.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
CustomView	Requis. Indique à quel objet CustomView l'objet SecurityObject s'applique. Type : vue personnalisée

## Type de retour

Aucun

## Exemples

```
new_namespace = model.Namespaces.Add()securityObject  
= new_namespace.SecurityObjects.Add()securityObject.AddToCustomView(customView)
```

---

## Méthode AddToFolder

La méthode AddToFolder ajoute une mesure à un dossier de mesures.

### Syntaxe

*mesure* .AddToFolder

### Application

Objet Measure

### Explication

En premier lieu, définissez la propriété IsFolder à vrai afin de créer un dossier de mesures. Utilisez la méthode AddToFolder pour ajouter une mesure à ce dossier.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requise. Spécifie l'objet de mesure à ajouter. Type : Objet - Measure

### Type de retour

Aucun

---

## Méthode AssociateWith

La méthode AssociateWith associe un objet source à un objet du modèle.

### Syntaxe

*object* .AssociateWith Reference, Role [, Context]

### Application

Objet CurrencyTable

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet Measure

## Explication

Utilisez cette méthode pour associer un objet de votre modèle à un objet d'une source de données. Le paramètre Référence exécute la même fonction que la propriété Reference d'un objet Association. Le paramètre Rôle exécute la même fonction que la propriété AssociationRole. Le paramètre Contexte exécute la même fonction que la propriété Context.

Quand vous utilisez la méthode AssociateWith, elle ajoute un objet Association à la collection Associations de l'objet. Par exemple, si la collection Associations d'un objet Measure possède déjà deux objets, la méthode AssociateWith lui ajoute un troisième objet.

La méthode AssociateWith est une alternative de la méthode Add de la collection Associations.

Quand vous utilisez la méthode AssociateWith avec un objet, continuez toujours avec la méthode Update du même objet.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Référence	Requise. Spécifie la source de l'association. Type : Objet
Rôle	Requise. Spécifie comment la référence est utilisée. Ce paramètre utilise une constante de la liste des valeurs de xtrObjectType. Type : Constante
Contexte	Requis pour les associations OrderBy. Spécifie la branche de niveaux dans laquelle un niveau est trié. S'applique aux objets de niveau et DateLevel uniquement. Type : Objet

## Type de retour

Objet - Association

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.
AssociateWith(objRefSource, xtrAssociationRole.trAssociationSource)
```

---

## Méthode CheckLocalPowerCubes

La méthode CheckLocalPowerCubes contrôle les cubes définis dans le modèle par rapport aux fichiers de PowerCube (.mdc) associés.

## Syntaxe

*Model* .CheckLocalPowerCubes

## Application

Objet Model

## Explication

Cette méthode permet de vérifier le statut des cubes du modèle et de les modifier si nécessaire.

Si le statut indique un avertissement, une erreur, une occupation ou OK, cette commande vérifie l'existence d'un fichier de PowerCube valide pour ce cube. Si Transformer n'arrive pas à localiser le fichier de PowerCube, il définit le statut du cube à Manquant. Si Transformer trouve le cube, mais ne peut pas l'ouvrir ou si le timbre de création du fichier ne correspond pas à celui du modèle (signifiant que le cube a été créé à partir d'un autre modèle ou d'une version obsolète du modèle en cours), il affecte le statut Erroné au PowerCube.

## Type de retour

Aucun

---

## Méthode CheckModel

La méthode CheckModel vérifie si le modèle en cours contient des objets en conflit avec la source de données, ou les uns avec les autres.

## Syntaxe

*Modèle* .CheckModel

## Application

Objet Model

## Explication

Transformer vérifie toujours le modèle avant de générer des catégories ou de créer des cubes, mais vous pouvez aussi le vérifier à tout moment à l'aide de cette méthode.

Nous vous conseillons de vérifier votre modèle aux points clés du processus de conception, notamment lors de l'ajout de nouvelles sources de données ou d'associations, ainsi qu'après avoir créé les définitions de cube.

La méthode CheckModel retourne une collection Names. Chaque objet Name de cette collection contient une chaîne de message complète. Vous pouvez utiliser la propriété Name pour lire les résultats d'une procédure CheckModel.

## Type de retour

Objet - Names

## Exemples

For intX = 1 To objModel.CheckModel.Count

---

## Méthode CleanHouse

La méthode CleanHouse supprime les catégories inactives d'un modèle ou d'une dimension.

### Syntaxe

*objet* .CleanHouse Date

### Application

Objet DateDimension

Objet Dimension

Objet Model

### Explication

Cette méthode permet de vérifier la présence de catégories inactives. Comme vos modèles sont adaptés aux modifications de votre entreprise, il arrive parfois qu'ils contiennent des catégories devenues inutiles. Une catégorie est considérée inactive si elle n'a pas été créée, mise à jour, déplacée ou modifiée depuis la date spécifiée par le paramètre Date.

Transformer vérifie la propriété LastUseDate de chaque objet Category pendant l'opération de nettoyage et la compare à la valeur du paramètre Date. Quand la propriété LastUseDate contient une date antérieure à la valeur du paramètre Date, Transformer supprime la catégorie.

Quand vous utilisez la méthode CleanHouse avec une dimension, Transformer ne vérifie que les catégories utilisées par cette dimension. Quand vous utilisez la méthode CleanHouse avec un modèle, Transformer vérifie toutes les catégories du modèle.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Date	Requise. Spécifie la date de suppression des catégories actives au format AAAAMMJJ.  Type : Long

### Type de retour

Aucun



---

## Méthode ClearDeployLocations

La méthode ClearDeployLocations supprime tous les emplacements de déploiement.

### Syntaxe

*Cube* .ClearDeployLocations

### Application

Objet Cube

### Explication

Cette méthode permet de supprimer tous les emplacements de déploiement de l'objet Cube. Cette méthode permet de préparer le déploiement d'un PowerCube ou de réaliser les opérations de nettoyage consécutives. Dans le cas d'un groupe de cubes, les cubes enfants sont également pris en compte. Cela ne s'applique qu'aux cubes découpés en partitions de temps.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Aucun

### Exemples

```
cube = model.Cubes.Item("National")cube.ClearDeployLocations()  
'Remove the deployment locations from the model
```

---

## Méthode Close

La méthode Close ferme le modèle en cours.

### Syntaxe

*Modèle* .Close

### Application

Objet Model

### Explication

Cette méthode permet de fermer le modèle en cours avant de créer ou d'ouvrir un autre modèle.

Si vous fermez un modèle avant de l'enregistrer, vous perdrez toutes les modifications que vous lui avez apportées, y compris les mises à jour des objets effectuées à l'aide de la méthode Update.

## Type de retour

Aucun

## Exemples

```
objModel.Close()
```

---

## Méthode ConnectWithCategory

La méthode ConnectWithCategory déplace une catégorie enfant vers une nouvelle catégorie parent dans le même niveau.

### Syntaxe

```
Catégorie .ConnectWithCategory(Parent [, DrillDown])
```

### Application

Objet Category

### Explication

Vous pouvez aussi utiliser les méthodes MoveToCategory et MoveToLevel pour déplacer des objets Category et SpecialCategory vers une nouvelle position ou une autre collection.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Parent	Requise. Spécifie la catégorie parent de la connexion.  Type : Objet
DrillDown	Facultatif. Spécifie un objet DateDrillDown ou DrillDown si le parent est présent dans plusieurs branches de niveaux. La valeur peut être une position d'index numérique, une chaîne spécifiant un nom d'objet ou un objet défini dans le script.  Type : Variante

## Type de retour

Aucun

## Exemples

```
objDimension = objModel.Dimensions("Retailers")objCategories  
= objDimension.DrillDowns(1).CategoriesobjParentCategory  
= objCategories(parent_category_index)objCategories  
= objDimension.DrillDowns(1).CategoriesobjCategory  
= objCategories(5)objChildCategory  
= objCategory.ChildCategories(1)objChildCategory.  
ConnectWithCategory(objParentCategory)
```

---

## Méthode CreateAlternateDrillDown

La méthode CreateAlternateDrillDown crée une branche de niveaux secondaire dans une dimension et convertit le niveau en niveau de convergence.

### Syntaxe

*objet* .CreateAlternateDrillDown

### Application

Objet DateLevel

Objet de niveau

### Explicatiion

Cette méthode permet de créer une branche de niveaux secondaire dans une dimension. Appliquez-la au niveau que vous souhaitez utiliser comme niveau de convergence. Transformer crée ensuite une branche secondaire de niveaux entre le premier niveau de la dimension et le niveau de convergence. Vous devez ensuite insérer un niveau dans la nouvelle branche pour établir la connexion au niveau de convergence et associer ce nouveau niveau à une valeur source. Comme Transformer ajoute toujours les objets à la fin de la collection, vous devez déplacer le nouveau niveau au sein de la collection Levels à une position qui précède le niveau de convergence.

Le type de branche de niveaux qui est créé, à savoir un objet DrillDown ou DateDrillDown, dépend du type de niveau. La nouvelle branche de niveaux est ajoutée à la collection DrillDowns de cette dimension.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Objet

### Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")objLevel  
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill  
= objLevel.CreateAlternateDrillDown
```

---

## Méthode CreateDateDimension

La méthode CreateDateDimension crée un objet DateDimension.

### Syntaxe

*DateWizard* .CreateDateDimension

### Application

Objet DateWizard

## Explication

Une fois que vous avez défini les propriétés applicables de l'objet DateWizard et que vous avez affecté une source de données avec la méthode DimensionAssociateWith, utilisez cette méthode pour créer un objet DateDimension.

Cette méthode retourne un objet DateDimension.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Objet - DateDimension

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.DimensionAssociateWith  
(objColumn, xtrAssociationRole.trAssociationSource)
```

---

## Méthode CreateMDCFile

La méthode CreateMDC File crée des fichiers PowerCube (.mdc) pour un cube unique ou pour tous les cubes d'un groupe de cubes.

## Syntaxe

*object* .CreateMDCFile

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette méthode pour créer un PowerCube à partir de la définition d'un cube ou d'un groupe de cubes de votre modèle.

Transformer génère automatiquement des catégories à partir des colonnes de vos sources de données quand vous créez un cube. Il inclut une catégorie dans le cube en fonction des paramètres de diverses propriétés, telles que Consolidate, Inclusion et DuplicateRollup.

Avant d'utiliser cette méthode, assurez-vous que les propriétés LocalPath et DataSourcePath sont correctement définies.

Utilisez la méthode CreateMDCFiles pour créer tous les cubes d'un modèle.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.CreateMDCFile()
```

---

## Méthode CreateMDCFile

La méthode CreateMDCFiles crée tous les fichiers de PowerCube (.mdc) dans un modèle.

### Syntaxe

*Model* .CreateMDCFiles

### Application

Objet Model

### Explication

Utilisez cette méthode pour créer un PowerCube pour chaque cube ou groupe de cubes d'un modèle.

Transformer génère automatiquement des catégories à partir des colonnes de vos sources de données quand vous créez un cube. Il inclut une catégorie dans le cube en fonction des paramètres de diverses propriétés, telles que Consolidate, Inclusion et DuplicateRollup.

Avant d'utiliser cette méthode, assurez-vous que les propriétés LocalPath et DataSourcePath sont correctement définies.

Pour créer des fichiers PowerCube (.mdc) pour un cube ou un groupe de cubes unique, utilisez la méthode CreateMDCFile.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

---

## Méthode Delete

La méthode Delete supprime un objet à partir d'un modèle.

### Syntaxe

*objet* .Delete()

## Application

La méthode Delete s'applique aux objets suivants :

Objet Association

Objet CalculationDefinition

Objet Category

Objet CategorySet

Objet Column

Objet CrossTabDataSource

«Objet Cube», à la page 61

Objet CubeGroup

Objet CurrencyRate

Objet CurrencyRecord

Objet CurrencyTable

Objet DateDimension

Objet DateDrillDown

Objet DataSource

Objet DateLevel

Objet DbDataSource

Objet Dimension

Objet DrillDown

Objet DrillThroughTarget

Objet Filter

Objet FlatFileDataSource

Objet IqdDataSource

Objet de niveau

Objet Measure

Objet Namespace

Objet PackageDatasourceConnection

Objet Package  
Objet Prompt  
Objet Query  
Objet Report  
Objet SecurityObject  
Objet de code d'accès  
Objet SpecialCategory  
Objet View

### **Explication**

Cette méthode permet de supprimer un objet du modèle quand il existe en dehors d'une collection. Pour enlever un objet d'une collection, utilisez la méthode Remove.

Vous ne pouvez pas supprimer la branche principale de niveaux créée par Transformer pour chaque dimension.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type de retour**

Aucun

### **Exemples**

```
objLevel  
= objTimeDimension.DrillDowns.Item(1).Levels.Item("Month").objLevel.Delete()
```

---

## **Méthode DeleteAllCustomViews**

La méthode DeleteAllCustomViews supprime toutes les vues personnalisées relatives à un modèle.

### **Syntaxe**

```
Modèle.DeleteAllCustomViews()
```

### **Application**

Objet Model

### **Explication**

Les objets de sécurité affectés sont également supprimés en même temps que les vues personnalisées.

## Type de retour

Aucun

---

## Méthode DeleteAllSecurityObjects

La méthode DeleteAllSecurityObjects supprime les objets de sécurité d'un modèle.

### Syntaxe

*Modèle* .DeleteAllSecurityObjects()

### Application

Objet Model

### Explication

Lorsque les vues personnalisées sont supprimées, les objets de sécurité affectés à la vue le sont également.

## Type de retour

Aucun

---

## Méthode DeployCube

La méthode DeployCube déploie un PowerCube vers tous les emplacements de déploiement.

### Syntaxe

*Cube* .DeployCube()

### Application

Objet Cube

### Explication

Cette méthode permet de déployer tous les PowerCubes. Appelez PublishDatasource ou PublishPackage après avoir déployé le PowerCube.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

### Exemples

```
cube = model.Cubes.Item("National")cube.DeployCube()
```



---

## Méthode DeployCubes

La méthode DeployCubes déploie tous les PowerCubes d'un modèle vers tous les emplacements de déploiement.

### Syntaxe

*Modèle* .DeployCubes()

### Application

Objet Model

### Explication

Cette méthode permet de déployer tous les PowerCubes pour un modèle. Appelez PublishDatasource ou PublishPackage après avoir déployé les PowerCubes.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Valeur booléenne

---

## Méthode DimensionAssociateWith

La méthode DimensionAssociateWith associe un objet source à un nouvel objet DateDimension et ajoute un objet Association à la collection Associations.

### Syntaxe

*DateWizard* .DimensionAssociateWith Référence, Rôle [, Contexte]

### Application

Objet DateWizard

### Explication

Utilisez cette méthode pour associer un objet DateDimension à un objet d'une source de données. Le paramètre Référence exécute la même fonction que la propriété Reference d'un objet Association. Le paramètre Rôle exécute la même fonction que la propriété AssociationRole. Le paramètre Contexte exécute la même fonction que la propriété Context.

Quand vous utilisez la méthode DimensionAssociateWith, elle ajoute un objet Association à la collection Associations de l'objet DateDimension.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Référence	Requise. Spécifie la source de l'association. Type : Objet
Rôle	Requise. Spécifie comment la référence est utilisée. Ce paramètre utilise une constante de la liste des valeurs de xtrObjectType. Type : Constante - xtrAssociationRole
Contexte	Requis pour les associations OrderBy. Spécifie la branche de niveaux dans laquelle un niveau est trié. S'applique aux objets de niveau et DateLevel uniquement. Type : Objet

## Type de retour

Aucun

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.DimensionAssociateWith
(objColumn, xtrAssociationRole.trAssociationSource)
```

---

## Méthode DoAutoDesign

La méthode DoAutoDesign génère des dimensions, des niveaux, des branches de niveaux, des cubes et des mesures.

## Syntaxe

*Modèle* .DoAutoDesign

## Application

Objet Model

## Explication

Cette méthode permet de générer les collections principales d'un nouveau modèle ou d'ajouter des dimensions et des niveaux à un modèle existant en fonction des colonnes qui n'ont pas été utilisées précédemment.

La méthode analyse les types de données, les noms de colonnes et le cadre structurel des sources de données. Elle place les dates dans une dimension de temps, les colonnes contenant des valeurs numériques dans la collection Measures et toutes les autres colonnes dans des objets Dimension. Les nouvelles dimensions contiennent des collections Levels et DrillDowns.

## Type de retour

Aucun

## Exemples

```
objModel = objTransApp.NewModel objModel.DoAutoDesign()
```

---

## Méthode FindCategoryByCatCode

La propriété FindCategoryByCatCode retourne l'objet Category contenant la chaîne du code de catégorie spécifiée.

### Syntaxe

*Dimension* .FindCategoryByCatCode CatCode

### Application

Objet Dimension

### Explication

Utilisez cette méthode pour retourner l'objet Category contenant le code de catégorie spécifié.

Si elle ne trouve pas de catégorie comportant ce code, la méthode FindCategoryByCatCode ne retourne aucune valeur. L'utilisation de cette méthode pour retourner une catégorie est beaucoup plus rapide que d'accéder à la catégorie via la collection Categories.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
CatCode	Requise. Indique la catégorie associée. <b>Type</b> : Chaîne

### Type de retour

Catégorie

---

## Méthode GenerateCategories

La méthode GenerateCategories alimente un modèle avec des catégories.

### Syntaxe

*Modèle* .GenerateCategories

### Application

Objet Model

### Explication

Cette méthode permet de générer des catégories à partir de votre source de données après l'utilisation de la méthode DoAutoDesign ou l'ajout manuel de dimensions et de niveaux à votre modèle. Pendant la génération des catégories, Transformer lit et analyse votre source de données et crée la hiérarchie des catégories. Les catégories sont générées automatiquement quand vous créez un cube.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

## Exemples

```
objModel.GenerateCategories()
```

---

## Méthode GenerateDateCategories

La méthode GenerateDateCategories alimente un modèle avec les catégories de date spécifiées par les paramètres EarliestDate et LatestDate.

## Syntaxe

```
DateDimension.GenerateDateCategories()
```

## Application

Objet DateDimension

## Explication

Utilisez cette méthode pour générer des catégories de date à partir de la source de données après avoir créé l'objet DateDimension et l'objet DrillDown associé avec les propriétés EarliestDate et LatestDate représentant la plage de catégories de date générées. Utilisez cette méthode après la méthode DoAutoDesign ou ajoutez manuellement des dimensions et des niveaux à votre modèle. Pendant la génération des catégories, Transformer lit et analyse votre source de données et crée la hiérarchie des catégories. Les catégories sont générées automatiquement quand vous créez un cube.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception est TR0821.

## Type de retour

Aucun

Paramètre	Description
EarliestDate	Requise. Spécifie la première catégorie de date. La valeur doit avoir le format de date AAAAMMJJ. S'il ne s'agit pas d'une valeur de date ou s'il s'agit d'une date antérieure à la dimension de date EarliestDate, le modèle enregistré sera incorrect.  Type : Long
LatestDate	Requise. Spécifie la dernière catégorie de date. La valeur doit avoir le format de date AAAAMMJJ.  Type : Long

## Exemples

```
objDateDim  
= objModel.Dimensions("Date")objDateDim.GenerateDateCategories(19990101, 20101231)
```

---

## Méthode GetDefaultCategory

La méthode GetDefaultCategory retourne la catégorie par défaut appartenant à une dimension. Cette catégorie est utilisée lorsque le rapport de type tableau croisé est ouvert pour la première fois.

### Syntaxe

*DateDimension* .GetDefaultCategory

### Application

Objet DateDimension

Objet Dimension

### Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Objet de catégorie

Paramètre	Description
Aucun	

---

## Méthode GetViewStatus

La méthode GetViewStatus retourne le statut d'affichage d'un objet Category, SpecialCategory, de niveau ou DateLevel.

### Syntaxe

*Vue* .GetViewStatus

### Application

Objet View

### Explication

Lorsque vous obtenez un objet View, vous pouvez l'utiliser pour extraire le statut d'affichage d'un objet Category, SpecialCategory, DateLevel ou de niveau associé. La constante retournée est une des valeurs possibles de la liste des valeurs de xtrViewStatus.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Constante - xtrViewStatus

Paramètre	Description
Objet	Requise. Cette méthode ne peut être appliquée qu'à un objet Category, SpecialCategory, DateLevel ou de niveau.  Type : Objet

## Exemples

```
objDimension = objModel.Dimensions("Retailers")objView  
= objDimension.Views.Add()If objView.GetViewStatus(objCategory)  
<> _xtrViewStatus.trViewStatusSummaryMom Then
```

---

## Méthode isExcludeDateDimension

La méthode isExcludeDateDimension retourne une valeur indiquant si une dimension de date particulière est exclue.

### Syntaxe

*DrillThroughTarget* .isExcludeDateDimension

### Application

Objet DrillThroughTarget

### Explication

Cette propriété permet d'extraire une valeur indiquant si la dimension de date spécifiée est exclue de la cible d'accès au détail.

Paramètre	Description
DateDimension	Requise. Indique à quelle dimension de date la propriété s'applique.

## Type de retour

Chaîne

### Accès

Lecture/écriture

---

## Méthode isExcludeDateLevel

La méthode isExcludeDateLevel retourne une valeur indiquant si un niveau de date particulier est exclu.

## Syntaxe

*DrillThroughTarget* .isExcludeDateLevel

## Application

Objet DrillThroughTarget

## Explication

Cette propriété permet d'extraire une valeur indiquant si le niveau de date spécifié est exclu de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
DateLevel	Requise. Indique à quel niveau de date la propriété s'applique. Type : DateLevel

## Type de retour

Valeur booléenne

## Accès

Lecture/écriture

---

## Méthode isExcludeDimension

La méthode isExcludeDimension retourne une valeur indiquant si une dimension est exclue.

## Syntaxe

*DrillThroughTarget* .isExcludeDimension

## Application

Objet DrillThroughTarget

## Explication

Cette propriété permet d'extraire une valeur indiquant si la dimension spécifiée est exclue de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Objet de niveau	Requise. Indique à quelle dimension la propriété s'applique. Type : Objet - Level

## Type

Chaîne

## Accès

Lecture/écriture

---

## Méthode isExcludeLevel

Pour un objet Report donné, la méthode isExcludeLevel spécifie si un objet de niveau est exclu.

## Syntaxe

*DrillThroughTarget* .isExcludeLevel

## Application

Objet DrillThroughTarget

## Explication

Cette propriété permet d'extraire une valeur indiquant si le niveau spécifié est exclu de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Niveau	Requise. Indique à quel niveau la propriété s'applique.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Méthode Item

La méthode Item retourne un objet spécifique dans une collection.



## Syntaxe

*collection* .Item(Key, [Type])

## Application

Collection Categories

Collection Cubes

Collection DataSources

Collection Dimensions

Collection DrillDowns

Collection Levels

Collection PackageDatasourceConnections

Collection Packages

Collection Queries

Collection Reports

## Explication

Le paramètre Type modifie le paramètre Clé lorsqu'une collection peut contenir plusieurs types d'objet. Si vous n'utilisez que le paramètre Clé, la méthode Item compte les objets depuis le début de la collection jusqu'à l'objet spécifié ou jusqu'à la fin de la collection. Si vous ajoutez le paramètre facultatif Type, la méthode Type ne comptabilise que ce type d'objet. De cette façon, les paramètres Clé et Type vous permettent de sélectionner un type d'objet spécifique à un certain endroit de la collection.

La collection Dimensions, par exemple, peut contenir à la fois des objets Dimension et DateDimension. Pour obtenir le cinquième objet de la collection, quel que soit son type, utilisez une instruction similaire à celle-ci. Notez que cet exemple s'applique à VB.NET, mais pas à C#. Avec C#, les deux paramètres doivent être spécifiés, que le type soit utilisé ou non.

```
Set objDim = objDimColl.Item(5)  
Dimension dim = (Dimension)model.Dimensions.Item(5,null);
```

Pour obtenir la troisième occurrence d'un objet DateDimension dans une collection d'objets Dimension et DateDimension, utilisez une instruction similaire à celle-ci :

```
Set objDim = objDimColl.Item(3, "trDateDimension")
```

Le paramètre Clé peut également être une chaîne désignant l'objet. Dans ce cas, utilisez une instruction similaire à celle-ci pour obtenir un objet DateDimension nommé Dates Comm :

```
Set objDim = objDimColl.Item("Order Dates")
```

Le paramètre Type ne s'applique qu'aux objets des collections Categories, Cubes, DataSources et Dimensions. Lors d'une itération sur l'ensemble des éléments d'une collection comprenant plusieurs types d'objet, la valeur nulle doit être affectée au paramètre Type.

La méthode Item peut être par défaut si le paramètre Type n'est pas utilisé. Ainsi, cette instruction (applicable uniquement à VB.NET) produit le même effet :

```
objModel.Dimensions("Dates").DrillDowns(1).Levels("Month")
```

**Remarque :** Depuis la version 7.0 du programme, la méthode Item est sensible à la casse.

Ainsi, l'exemple suivant n'établit de correspondance qu'avec une dimension appelée "Ligne" et non "ligne" :

```
objModel.Dimension("Line")
```

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Clé	Requise. Indique une position d'index numérique commençant à 1 ou une chaîne désignant le nom de l'objet à récupérer.  <b>Type :</b> Variante
Type	Facultatif. Spécifie une constante de la liste des valeurs de xtrObjectType. Ce paramètre s'applique uniquement aux collections Categories, Cubes, DataSources, Dimensions, DimensionLevels, DrillDowns et Levels.  <b>Type :</b> Constante

## Type de retour

Objet

## Exemples

```
objDimensions = objModel.DimensionsobjLocationsDim  
= objDimensions.Item(3)objLocationsDim  
= objDimensions.Item(3)
```

---

## Méthode Item ()

La méthode Item retourne un objet spécifique dans une collection.

### Syntaxe

*collection* .Item(Clé)

### Application

Collection Associations

Collection CalculationDefinitions

Collection CategorySets  
 Collection ChildCubes  
 Collection Columns  
 Collection CubeCustomViews  
 Collection Cubes  
 Collection CurrencyRates  
 Collection CurrencyRecords  
 Collection CurrencyTables  
 collection CustomViews  
 Collection DrillThroughTargets  
 Collection Filters  
 Collection LevelDrillDowns  
 Collection Measures  
 Collection Names  
 Collection Namespaces  
 Collection Prompts  
 Collection SecurityObjects  
 Collection Signons  
 Collection SuspendedModels  
 Collection Views

### Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Clé	Requise. Indique une position d'index numérique commençant à 1 ou une chaîne désignant le nom de l'objet à récupérer. Une chaîne désignant le nom de l'objet à récupérer ne peut pas être utilisée pour les associations.  <b>Type</b> : Variante

## Type de retour

Objet

---

## Méthode Item ()

La méthode Item retourne un objet spécifique dans une collection.

### Syntaxe

*collection* .Item(Clé, Type)

### Application

Collection DimensionLevels

collection LevelCategories

### Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Clé	Requise. Indique une position d'index numérique commençant à 1 ou une chaîne désignant le nom de l'objet à récupérer. Une chaîne désignant le nom de l'objet à récupérer ne peut pas être utilisée pour les associations.  <b>Type</b> : Variante
Type	Requise. Spécifie une constante de la liste des valeurs de xtrObjectType. Ce paramètre s'applique uniquement aux collections Categories, Cubes, DataSources, Dimensions, DimensionLevels, DrillDowns et Levels. Avec certains langages d'implémentation, si cette valeur n'est pas requise, elle doit être définie sur nulle, comme indiqué ci-dessus.  <b>Type</b> : Constante

## Type de retour

Objet

---

## Méthode LoadCurrencyTable

La méthode LoadCurrencyTable charge une table des devises dans le modèle.

### Syntaxe

*Modèle* .LoadCurrencyTable

## Application

Objet Model

## Explication

Utilisez cette méthode pour charger une table des devises avec les informations d'une source de données externe. Vous ne pouvez utiliser cette méthode que si la collection CurrencyTables contient au moins une table des devises.

Si vous ajoutez une table des devises en euro à la collection CurrencyTables, vous devez utiliser la méthode LoadCurrencyTable à différentes étapes du processus. Utilisez cette méthode après avoir associé chaque colonne de la source de données à un rôle d'association de la table en euro et après avoir ajouté les enregistrements de devise à la table en euro.

## Type de retour

Aucun

## Exemples

```
objModel.LoadCurrencyTable()
```

---

## Méthode Logoff

La méthode Logoff permet la déconnexion de tous les espaces-noms.

## Syntaxe

*Application* .Logoff()

## Application

Objet Application

## Explication

Cette propriété permet une déconnexion des espaces-noms en cours. Utiliser avec la méthode Logoff.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

---

## Méthode Logon

La méthode Logon permet la connexion à un espace-noms.

## Syntaxe

*Application* .Logon(EspaceNoms, NomUtilisateur, MotDePasse)

## Application

Objet Application

## Explication

Cette propriété permet une connexion à un espace-noms. Utiliser avec la méthode Logoff.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Espace-noms	Requise. Spécifie un espace-noms pour la connexion. Type : chaîne
Nom d'utilisateur	Requise. Spécifie un nom d'utilisateur pour la connexion. Type : chaîne
Mot de passe	Requise. Spécifie un mot de passe correspondant au nom d'utilisateur. Type : chaîne

## Type de retour

Aucun

## Exemples

```
objTransApp.Logon("Cognos", "", "") 'Log into the Cognos namespace using Anonymous user.
```

---

## Méthode Move

La méthode Move réorganise les objets d'une collection.

## Syntaxe

```
objet .Move(Avant, Après)
```

## Application

«Objet CrossTabDataSource», à la page 59

Objet DataSource

Objet DateDimension

Objet DateLevel

«Objet DbDataSource», à la page 85

Objet Dimension

Objet FlatFileDataSource

«Objet IqdDataSource», à la page 95

Objet de niveau

Objet Measure

## Explication

Les paramètres Avant et Après s'excluent mutuellement. Vous devez utiliser un seul paramètre. Si vous omettez ces deux paramètres, une erreur se produit et une exception COM est générée. Si les deux paramètres sont utilisés, seul le paramètre Avant est pris en compte. Si des objets erronés sont utilisés, une exception COM est générée.

Utilisez la méthode Item pour sélectionner l'objet dans une collection.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Avant	Requis si le paramètre Après n'est pas utilisé. La valeur peut correspondre à une position d'index numérique débutant à 1, ou à un objet défini dans le script.  Type : Variante
Après	Requis si le paramètre Avant n'est pas utilisé. La valeur peut correspondre à une position d'index numérique débutant à 1, ou à un objet défini dans le script.  Type : Variante

## Type de retour

Aucun

## Exemples

```
objMeasures = objModel.MeasuresobjMeasures.Item(2).Move(1)
```

---

## Méthode MoveToCategory

La méthode MoveToCategory déplace une catégorie enfant vers une autre catégorie parent.

## Syntaxe

```
Category .MoveToCategory(Parent [, Sibling])
```

## Application

Objet Category

## Explication

Vous pouvez déplacer la catégorie à n'importe quel niveau situé sous le nouveau parent. Le paramètre Soeur vous permet de positionner la catégorie que vous déplacez au sein de la liste des catégories existantes à la nouvelle position.

Vous pouvez aussi déplacer des catégories vers une nouvelle position à l'aide des méthodes ConnectWithCategory et MoveToLevel.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur. Voici quelques exemples :

- "Le serveur a généré une exception. (Exception HRESULT: 0x80010105 (RPC\_E\_SERVERFAULT))"
- "trCategory(Category) : Objet de type inattendu" (si l'un des paramètres possède un type inattendu)

Paramètre	Description
Parent	Requise. Spécifie la catégorie parent de la connexion.  Type : Objet
Soeur	Facultatif. Spécifie une catégorie enfant au même niveau pour positionner la catégorie déplacée après celle-ci. Si ce paramètre n'est pas spécifié, la catégorie est déplacée à la fin de la liste des catégories enfants. La valeur peut être une position d'index numérique, une chaîne spécifiant un nom d'objet ou un objet défini dans le script.  Type : Variante

## Type de retour

Aucun

## Exemples

```
objCategory  
= objModel.Dimensions("Retailers").DrillDowns(1).Categories(2)objCatToMove  
= objCategory.ChildCategories(5)objCatToReceive  
= objModel.Dimensions("Retailers").DrillDowns(1).Categories(1)objCatToMove.  
MoveToCategory(objCatToReceive)
```

---

## Méthode MoveToLevel

La méthode MoveToLevel déplace une catégorie enfant vers un autre niveau situé sous la catégorie parent en cours.

## Syntaxe

*Catégorie* .MoveToLevel(Niveau)



## Application

Objet Category

## Explication

Vous pouvez déplacer la catégorie vers n'importe quel niveau supérieur ou inférieur, à condition que la nouvelle position soit située sous le parent d'origine.

Vous pouvez aussi déplacer des catégories vers une nouvelle position à l'aide des méthodes `ConnectWithCategory` et `MoveToCategory`.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Niveau	Requise. Spécifie le niveau dans lequel la catégorie doit être déplacée. Type : Objet

## Type de retour

Aucun

## Exemple

Aucun exemple n'est disponible.

---

## Méthode `NewModel`

La méthode `NewModel` crée un modèle.

## Syntaxe

*Application* .`NewModel`

## Application

Objet Application

## Explication

Cette méthode permet de créer un modèle.

Au moment de sa création, le modèle est généralement vide. Vous devez définir les sources de données, les dimensions, les niveaux et les mesures avant de pouvoir créer des cubes.

Pour les nouveaux modèles, enregistrez vos définitions à l'aide de la méthode `SaveAs` plutôt que la méthode `Save`.

Pour modifier un modèle existant, utilisez la méthode `OpenModel`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Modèle

## Exemples

```
objModel = objTransApp.NewModel
```

---

## Méthode OpenModel

La méthode OpenModel ouvre un modèle existant.

### Syntaxe

*Application* .OpenModel(Nom)

### Application

Objet Application

### Explication

Utilisez cette méthode pour référencer un modèle existant quand vous devez le modifier.

Pour créer un modèle, utilisez la méthode NewModel. Pour enregistrer les modifications apportées à un modèle, utilisez la méthode Save ou SaveAs. Pour fermer un modèle, utilisez la méthode Close.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Nom	Requis. Spécifie le nom de fichier complet du modèle. Type : chaîne
Connexion	Requis pour les modèles sécurisés. Spécifie l'ID utilisateur de la personne qui accède au modèle. Type : chaîne
Mot de passe	Requis pour les modèles sécurisés. Spécifie le mot de passe de la personne qui accède au modèle. Type : chaîne
UserClass	Requis pour les modèles sécurisés. Spécifie la classe d'utilisateurs de la personne qui accède au modèle. Type : chaîne

## Type de retour

Objet

## Exemples

```
objModel = objTransApp.OpenModel(strModelPath)
```

---

## Méthode OpenSuspendedModel

La méthode OpenSuspendedModel ouvre un modèle en attente existant.

### Syntaxe

```
Application .OpenSuspendedModel(Name)]
```

### Application

Objet Application

### Explication

Utilisez cette méthode pour ouvrir un modèle incomplet existant.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Objet SuspendedModel	Requise. Représente un modèle incomplet existant  <b>Type</b> : Objet - Suspended Model

## Type de retour

Objet - Model

---

## Méthode PublishDatasource

La méthode PublishDatasource publie la connexion à la source de données pour un PowerCube.

### Syntaxe

```
Cube .PublishDatasource(overwrite)
```

### Application

Objet Cube

### Explication

Cette méthode permet de publier une connexion à une source de données sur le serveur IBM Cognos.

Vous devez indiquer un emplacement Windows, UNIX ou Linux à partir duquel le serveur IBM Cognos accède à chaque cube. Vous devez également indiquer l'espace-noms configuré si l'authentification utilisateur est activée. La valeur True est retournée en cas de réussite.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
remplacement	Requise. Associée à la valeur True, republie la source de données si elle existe déjà.  <b>Type</b> : Booléen

### Type de retour

Booléen (True si la publication aboutit)

### Exemples

```
cube = model.Cubes.Item("National").cube.PublishDatasource(True)
'Republish PowerCube if it already exists
```

---

## Méthode PublishPackage

La méthode PublishPackage publie la source de données et le pack pour un PowerCube.

### Syntaxe

```
Cube .PublishPackage(datasourceOverwrite, packageOverwrite)
```

### Application

Objet Cube

### Explication

Cette méthode permet de publier une connexion à une source de données et un pack sur le serveur IBM Cognos BI pour un PowerCube.

Vous devez indiquer un emplacement Windows, UNIX ou Linux à partir duquel le serveur IBM Cognos accède à chaque cube. Vous devez également indiquer l'espace-noms configuré si l'authentification utilisateur est activée. La valeur True est retournée en cas de réussite.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
datasourceOverwrite	Requise. Associée à la valeur True, republie la source de données si elle existe déjà. <b>Type</b> : Booléen
packageOverwrite	Requise. Si valeur True, republie le pack s'il existe déjà. <b>Type</b> : Booléen

## Type de retour

Booléen (True si la publication aboutit)

## Exemples

```
cube = model.Cubes.Item("National").cube.PublishPackage(True,
True) 'Re-publish both Datasource & Package
```

---

## Méthode Remove

La méthode Remove supprime un objet particulier d'une collection.

### Syntaxe

```
collection .Remove(Elément)
```

### Application

Collection Associations

Collection CalculationDefinitions

Collection Categories

Collection CategorySets

Collection Columns

Collection CubeCustomViews

Collection Cubes

Collection CurrencyRates

Collection CurrencyRecords

Collection CurrencyTables

collection CustomViews

Collection DataSources

Collection Dimensions

Collection DrillDowns

Collection DrillThroughTargets

Collection Filters

Collection Levels

Collection Measures

Collection Namespaces

Collection PackageDatasourceConnections

Collection Packages

Collection Prompts

Collection Queries

Collection Reports

Collection SecurityObjects

Collection Signons

Collection Views

## Explication

Utilisez le paramètre Elément de cette méthode pour sélectionner l'objet spécifique à supprimer de la collection. Utilisez la méthode Delete pour supprimer des objets qui existent en dehors d'une collection, notamment ceux qui sont sélectionnés avec la méthode Item.

Vous ne pouvez pas supprimer la branche principale de niveaux créée par Transformer pour chaque dimension.

Lors de la suppression d'associations, une mise à jour de l'objet parent est demandée. Par conséquent, toutes les modifications apportées à l'objet parent sont sauvegardées.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Elément	Requise. Spécifie un objet ou une position dans une collection. La valeur peut être une position d'index numérique (commençant à 1), une chaîne spécifiant un nom d'objet ou un objet défini dans le script.  Type : Variante

## Type de retour

Aucun

## Exemples

```
packages = model.Packages packages.Remove(temp_package)
```

---

## Méthode RemoveCubeCustomView

La méthode RemoveCubeCustomView supprime la vue personnalisée du cube ou du cube enfant.

### Syntaxe

*Object* .RemoveCubeCustomView

### Application

- Objet ChildCube
- Objet Cube

### Explication

Cette méthode permet de supprimer dans un cube ou un cube enfant une vue personnalisée du cube.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
CubeCustomView	Requise. Spécifie la vue personnalisée à supprimer. Type : vue personnalisée

## Type de retour

Aucun

---

## Méthode RemoveFromFolder

La méthode RemoveFromFolder supprime une mesure enfant du dossier des mesures actif.

### Syntaxe

*Mesure* .RemoveFromFolder Elément

### Application

Objet Measure

## Explication

Utilisez cette méthode pour supprimer des mesures enfant d'un dossier de mesures. Si la mesure enfant existe, elle sera supprimée au même niveau que le dossier de mesures. Si elle n'existe pas sous le dossier des mesures, l'application transmet une exception.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requise. Spécifie l'objet de mesure à supprimer du dossier des mesures. Type : Objet - Measure

## Type de retour

Aucun

---

## Méthode RemoveSuspendedModel

La méthode RemoveSuspendedModel supprime un modèle en attente de la collection SuspendedModels.

## Syntaxe

*Application* .RemoveSuspendedModel (*Modèle*)

## Application

Objet Application

## Explication

Utilisez la propriété IsBad pour vérifier si le modèle en attente est altéré ou récupérable. S'il est altéré, utilisez la méthode RemoveSuspendedModel pour le supprimer de la collection SuspendedModels.

Paramètre	Description
Modèle	Requise. Représente un modèle incomplet existant. Type : SuspendedModel

## Type de retour

Booléen

## Exemples

```
objTransApp.RemoveSuspendedModel(objSuspendedModel)
```



---

## Méthode ResetPartitions

La méthode ResetPartitions supprime les partitions du cube en cours.

### Syntaxe

*Modèle* .ResetPartitions

### Application

Objet Model

### Explication

Si vos cubes sont extrêmement volumineux ou structurés de manière inhabituelle, vous pouvez réduire leur temps de création et améliorer leurs performances d'exécution en définissant manuellement vos partitions. Utilisez d'abord cette méthode pour supprimer toutes les partitions manuelles ou automatiques qui ont été préalablement définies. Utilisez ensuite la propriété Partition pour affecter des numéros de niveau de partition aux catégories dans les dimensions sélectionnées.

### Type de retour

Objet

### Exemples

```
objModel.ResetPartitions()
```

---

## Méthode Save

La méthode Save enregistre les modifications apportées au modèle en cours.

### Syntaxe

*Modèle* .Save

### Application

Objet Model

### Explication

Utilisez cette méthode pour écrire les modifications que vous avez apportées au modèle dans un fichier modèle (.mdl). La méthode SaveAs permet d'enregistrer les modifications sous un autre nom de fichier et d'enregistrer pour la première fois un nouveau modèle. Si vous n'utilisez pas la méthode Save ou SaveAs avant de fermer un modèle, vous perdez toutes les modifications apportées au modèle, y compris celles appliquées à l'aide de la méthode Update.

### Type de retour

Aucun

---

## Méthode SaveAs

La méthode SaveAs enregistre le modèle en cours sous un autre nom de fichier.

## Syntaxe

Modèle `.SaveAs` NomFichier

## Application

Objet Model

## Explication

Utilisez cette méthode pour écrire les modifications que vous avez apportées au modèle dans un autre fichier modèle (.mdl). Utilisez toujours la méthode `SaveAs` quand vous enregistrez un nouveau modèle pour la première fois.

Utilisez la méthode `Save` pour enregistrer les modifications sous le même nom de fichier. Si vous n'utilisez pas la méthode `Save` ou `SaveAs` avant de fermer un modèle, vous perdez toutes les modifications apportées au modèle, y compris celles appliquées à l'aide de la méthode `Update`.

Paramètre	Description
FileName	Requise. Fournit un nom de fichier et éventuellement, un chemin d'accès.  Type : chaîne  Si aucun chemin d'accès n'est spécifié avec le nom de fichier, Transformer utilise le répertoire par défaut.

## Type de retour

Aucun

## Exemples

```
objModel.SaveAs("great outdoors salesX.mdl")
```

---

## Méthode SetAllocation

La méthode `SetAllocation` modifie le type d'allocation d'une mesure utilisée par un objet.

## Syntaxe

*object* `.SetAllocation` Measure, AllocationType [, AllocationMeasure ]

## Application

«Objet Category», à la page 50

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet SpecialCategory

## Explication

L'allocation n'est possible que lorsque votre modèle utilise plusieurs sources de données. Cette méthode permet alors de spécifier si Transformer alloue les données récapitulatives dans une dimension, un niveau ou une catégorie, ainsi que le type d'allocation.

Lorsque vous modifiez le type d'allocation d'un niveau, il s'applique à partir des catégories de ce niveau jusqu'à toutes les catégories descendantes.

Quand la source d'une mesure est une source de données qui ne référence pas une dimension dans le modèle, Transformer alloue automatiquement la mesure à la totalité de la dimension, puis la valeur de mesure en tant que constante dans la dimension. Pour rappeler cette allocation, utilisez la méthode SetAllocation avec la constante trAllocationNA.

La propriété CanAllocate doit retourner la valeur True pour poursuivre l'allocation.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requise. Spécifie la mesure à laquelle est appliquée l'allocation. Type : Object-Mesure
AllocationType	Requise. Définit l'option d'allocation. Spécifie une constante de la liste des valeurs de xtrAllocationType. Spécifie l'objet à utiliser comme source des valeurs d'allocation. Type : Constante - xtrAllocationType
AllocationMeasure	Facultatif. S'applique uniquement quand le paramètre TypeAllocation a pour valeur trAllocationAllocated. Type : Objet

## Type de retour

Aucun

## Exemples

```
objDimension.SetAllocation(objMeasure, xtrAllocationType.trAllocationNA)
```

---

## Méthode SetDefaultCategory

La méthode SetDefaultCategory spécifie une nouvelle catégorie par défaut pour une dimension, différente de la catégorie racine ou parent d'un groupe de catégories de scénarios. Pour chaque dimension, vous ne pouvez indiquer qu'une seule catégorie par défaut.

## Syntaxe

*Dimension* .SetDefaultCategory

## Application

- Objet DateDimension
- Objet Dimension

## Explication

Pour créer une dimension, nous vous recommandons d'utiliser cette méthode avec la propriété HideValue de l'objet Category racine ou parent.

En définissant un niveau par défaut pour ouvrir le cube, vous garantissez que les cubes contenant des valeurs de budget ou d'autres données de type scénario n'affichent aucun zéro, aucun nombre non applicable, ni nombre non significatif à leur ouverture par les auteurs ou utilisateurs d'un rapport.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun.

Paramètre	Description
Catégorie	Requise. Dans la dimension donnée, spécifie un objet Category valide.  Type : Objet - Category

---

## Méthode SetDeployType

La méthode SetDeployType définit la stratégie de déploiement pour la copie et l'activation.

## Syntaxe

*Cube* .SetDeployType(deployType

## Application

Objet Cube

## Explication

Utilisez cette méthode avant de déployer un PowerCube.

Le paramètre deployType permet à un utilisateur de définir une stratégie de déploiement ; trDeployType\_NONE spécifie qu'il est impossible de déployer un PowerCube.

Une valeur trDeployType\_SWAPSINGLE spécifie que si un ou plusieurs emplacements de déploiement sont indisponibles, l'action de déploiement est abandonnée pour l'ensemble des emplacements définis.

Une valeur `trDeployType_SWAPTOGETHER` spécifie que si un ou plusieurs emplacements définis en tant qu'emplacements de déploiement (voir `AddDeployLocation`) sont indisponibles, le `PowerCube` n'est pas déployé sur ces emplacements. Toutefois, le déploiement n'a pas lieu sur tous les emplacements disponibles.

Pour toutes les valeurs hormis `trDeployType_NONE`, le ou les emplacements de déploiement doivent être définis avant l'appel de la méthode `DeployCube`.

S'il s'agit d'un cube enfant appartenant à un groupe de cubes, le cube enfant peut hériter de la même stratégie de déploiement que son cube parent. Cela ne s'applique pas aux cubes découpés en partitions de temps.

Par défaut, la stratégie de déploiement est "Ne pas déployer".

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type de retour

Aucun

Paramètre	Description
<code>deployType</code>	Requise. Spécifie le type de déploiement. Ce paramètre utilise une constante de la liste des valeurs de <code>xtrDeployType</code> .  Type : Constante - <code>xtrDeploytype</code>

## Exemples

```
cube
= model.Cubes.Item("National")cube.
    SetDeployType(xtrDeployType.trDeployType_SWAPSINGLE)
```

---

## Méthode `SetExcludeDateDimension`

Pour un objet `DrillThroughTarget` et `DateDimension` donné, la méthode `SetExcludeDateDimension` définit la limite d'accès au détail sur `True` si la valeur du paramètre `isRestricted` est `True`. Pour supprimer la limite, définissez le paramètre `isRestricted` sur `False`.

### Syntaxe

```
DrillThroughTarget .SetExcludeDateDimension
```

### Application

Objet `DrillThroughTarget`

### Explication

Utilisez cette propriété pour exclure `DateDimension` de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Dimension de dates :	Requise. Indique que l'objet de dimension de dates est exclu de l'action d'accès au détail. Type : DateDimension
IsRestricted	Requise. Indique si l'objet Date Dimension est exclu. Type : Booléen

## Type

Booléen

## Accès

Ecriture

---

## Méthode SetExcludeDateLevel

Pour un objet DrillThroughTarget et DateLevel donné, la méthode SetExcludeDateLevel définit la limite d'accès au détail sur True si la valeur du paramètre isRestricted est True. Pour supprimer la limite, définissez le paramètre isRestricted sur False.

## Syntaxe

*DrillThroughTarget* .SetExcludeDateLevel

## Application

Objet DrillThroughTarget

## Explication

Utilisez cette propriété pour exclure DateLevel de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Niveau de date	Requise. Indique que l'objet de niveau de date est exclu de l'action d'accès au détail. Type : Objet - DateLevel
IsRestricted	Requise. Indique si l'objet de niveau de date est exclu ou non. Type : Booléen

## Type

Booléen

## Accès

Ecriture

---

## Méthode SetExcludeDimension

Pour un objet *DrillThroughTarget* et *Dimension* donné, la méthode *SetExcludeDimension* définit la limite d'accès au détail sur *True* si la valeur du paramètre *isRestricted* est *True*. Pour supprimer la limite, définissez le paramètre *isRestricted* sur *False*.

## Syntaxe

*DrillThroughTarget* .**SetExcludeDateDimension**

## Application

Objet *DrillThroughTarget*

## Explication

Utilisez cette propriété pour exclure *Dimension* de la cible d'accès au détail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
<i>Dimension</i>	Requise. Indique que l'objet de dimension est exclu de l'action d'accès au détail. Type : <i>Dimension</i>
<i>isRestricted</i>	Requise. Indique si la dimension est exclue. Type : Booléen

## Type de retour

Aucun

---

## Méthode SetExcludeLevel

Pour un objet *DrillThroughTarget* et *Level* donné, la méthode *SetExcludeLevel* définit la limite d'accès au détail sur *True* si la valeur du paramètre *isRestricted* est *True*. Pour supprimer la limite, définissez le paramètre *isRestricted* sur *False*.

## Syntaxe

*DrillThroughTarget* .**SetExcludeLevel**

## Application

Objet DrillThroughTarget

## Explication

Utilisez cette propriété pour exclure le niveau de l'objet DrillThroughTarget.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Niveau	Requis. Indique que l'objet de niveau est exclu de l'action d'accès au détail. Type : Objet
isRestricted	Obligatoire. Indique si le niveau est exclu. Type : Booléen

## Type de retour

Booléen

---

## Méthode SetViewStatus

La méthode SetViewStatus définit de quelle façon un objet Category, SpecialCategory, de niveau ou DateLevel est affiché.

## Syntaxe

*Vue* .SetViewStatus *Objet*, *Statut*

## Application

Objet Vue

## Explication

Lorsque vous obtenez un objet View, vous pouvez l'utiliser pour définir le statut d'affichage (propriété ViewStatus) d'un objet Category, SpecialCategory, DateLevel ou de niveau associé. Affectez une constante de la liste des valeurs de xtrViewStatus pour définir la vue.

Pour créer une vue d'une catégorie et des catégories enfants, utilisez la propriété Apex.

Pour effacer l'état de la vue en cours, extrayez le statut en cours et appelez SetViewStatus avec le résultat. Cette opération bascule l'état en cours et l'efface.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



Paramètre	Description
Objet	Requise. Cette méthode ne peut être appliquée qu'à un objet Category, SpecialCategory, DateLevel ou de niveau. Type : Objet

## Type de retour

Constante - xtrViewStatus

## Exemples

```
objDimension = objModel.Dimensions("Retailers")objView
= objDimension.Views.Add()objView.SetViewStatus(objCategory,
xtrViewStatus.trViewStatusSummaryMom)
```

L'exemple suivant illustre la façon d'effacer l'état de la vue en cours.

```
xtrViewStatus currentStatus = objView.GetViewStatus(objCategory);
objView.SetViewStatus(objCategory, currentStatus);
```

---

## Méthode TestBuild

La méthode TestBuild crée un petit modèle ou cube de test.

### Syntaxe

*Model* .TestBuild Enregistrements, CréerCubes

### Application

Objet Model

### Explication

Utilisez cette méthode pour générer des catégories ou créer un cube à des fins de test. Elle permet de gagner un temps considérable lorsque la source de données est extrêmement volumineuse.

Vous pouvez, par exemple, spécifier un jeu d'essai qui utilise les 350 premiers enregistrements de chaque source de données pour créer un cube. Transformer crée rapidement un petit cube dont vous pouvez évaluer les catégories dans PowerPlay.

Paramètre	Description
Enregistrements	Requise. Spécifie le nombre d'enregistrements à tester. Type : Long

Paramètre	Description
CréerCubes	Requise. Spécifie s'il faut seulement ajouter des catégories ou les ajouter, puis créer un cube. Si la valeur est True, Transformer crée un cube.  Type : Booléen

## Type de retour

Aucun

## Exemples

```
objModel = objTransApp.NewModel objModel.TestBuild(20, True)
```

---

## Méthode Update

La méthode Update met à jour l'objet associé.

### Syntaxe

*object* .Update

### Application

Objet Association

Objet CalculationDefinition

Objet Category

Objet CategorySet

Objet ChildCube

Objet Column

Objet CrossTabDataSource

Objet Cube

Objet CubeGroup

Objet CurrencyRate

Objet CurrencyRecord

Objet CurrencyTable

Objet CustomView

Objet DataSource

Objet DateDimension  
Objet DateDrillDown  
Objet DateLevel  
Objet DbDataSource  
Objet Dimension  
Objet DrillDown  
Objet DrillThroughTarget  
Objet Filter  
Objet FlatFileDataSource  
Objet IqdDataSource  
Objet de niveau  
Objet Measure  
Objet Model  
Objet Namespace  
Objet Package  
Objet PackageDatasourceConnection  
Objet Prompt  
Objet Query  
Objet Report  
Objet SecurityObject  
Objet de code d'accès  
Objet SpecialCategory  
Objet View

## **Explication**

Cette méthode permet de définir les modifications apportées aux propriétés d'un objet. Utilisez la méthode Save ou SaveAs pour enregistrer les modifications apportées au modèle avant de le fermer, sous peine de perdre toutes les mises à jour des objets. Si vous n'utilisez pas la méthode Update, les modifications apportées à l'objet ne sont pas enregistrées avec le modèle.

La méthode Update n'est pas requise pour définir les modifications apportées à l'objet Application ou DateWizard.

Quand vous exécutez Update, Transformer vérifie les modifications apportées à l'objet pour déterminer si elles sont valides. Par exemple, si vous définissez une expression erronée dans la propriété ExpressionText d'un objet Category, la mise à jour échoue et Transformer émet des messages d'erreur.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Aucun

### Exemples

```
objDimensions = objModel.DimensionsobjLocationsDim  
= objDimensions.Item(3)objLocationsDim.Update()
```

---

## Méthode Verify

La méthode Verify vérifie l'objet associé.

### Syntaxe

*objet* .Verify()

### Application

Objet Package

Objet PackageDatasourceConnection

Objet Query

Objet Report

Objet SecurityObject

### Explication

Cette méthode permet de vérifier les modifications apportées aux propriétés d'un objet. La méthode Update vérifie aussi l'objet de sorte que cette méthode n'est pas requise si une mise à jour est appelée sur l'objet. Transformer vérifie les modifications apportées à l'objet pour déterminer si elles sont valides.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type de retour

Aucun

---

## Chapitre 6. Propriétés

Les tableaux ci-dessous récapitulent toutes les propriétés d'automatisation OLE de Transformer.

Propriété	Description
Propriété ActivityMeasure	Définit ou retourne la mesure utilisée par une mesure de nombre de catégories.
Propriété Aggregate	Définit ou retourne le type de base de catégorie de date relative.
Propriété AllocationMeasure	Retourne l'objet Measure utilisé comme facteur de pondération.
Propriété AllocationType	Retourne une valeur indiquant de quelle manière un objet alloue une mesure.
Propriété AllowCurrencyConversion	Définit ou retourne une valeur indiquant si vous pouvez modifier une devise.
Propriété AllowDrillThrough	Définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.
Propriété AlternateQueryPath	Définit ou retourne un chemin d'accès secondaire de la source de données utilisé par le cube.
Propriété AltMDCFile	Indique un autre nom de fichier pour le cube.
Propriété AlwaysUseTransformerSignon	Indique s'il faut utiliser le code d'accès Transformer ou Content Manager.
Propriété Apex	Définit ou retourne l'objet Category utilisé comme racine de la vue d'une dimension.
Propriété Application	Retourne l'objet Application Transformer.
Propriété AssociationRole	Définit ou retourne le rôle de l'objet Association.
Propriété Associations	Retourne une collection Associations.
Propriété AssociationType	Définit ou retourne le type de la source de données associée à un objet Association.

Propriété	Description
Propriété AutoLogon	Si la valeur est True, indique si l'application s'authentifie automatiquement auprès des espaces-noms sécurisés associés au code d'accès.
Propriété AutoSummary	Définit ou retourne l'option AutoSummary pour une requête.
Propriété BlankSubstitute	Définit ou retourne le libellé par défaut des catégories en blanc générées dans le niveau de date.
Propriété BlockParentTotals	Définit ou retourne une valeur indiquant si des parents de catégories exclues affichent une valeur refusée.
Propriété CacheCrossTabs	Définit ou retourne une valeur indiquant si les récapitulatifs sont stockés dans le cache pour le tableau PowerPlay initial du cube.
Propriété CalculationDefinitions	Retourne une collection d'objets CalculationDefinition.
Propriété CAMID	Renvoie le CAMID de l'espace-noms.
Propriété CanAllocate	Retourne une valeur indiquant si vous pouvez allouer n'importe quelles valeurs de mesure aux niveaux descendants et aux catégories.
Propriété CanAllocateByMeasure	Retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.
Propriété CanAllocateMeasure	Retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.
Propriété Categories	Retourne une collection Categories associée à un objet.
Propriété Category	Retourne la catégorie de date applicable pour un taux de conversion donné.
Propriété CategoryCount	Retourne le nombre de catégories ordinaires dans une dimension ou un niveau.
Propriété CategoryCountLevel	Définit ou retourne l'objet de niveau auquel s'applique une mesure de nombre de catégories.

<b>Propriété</b>	<b>Description</b>
Propriété CategorySets	Retourne la collection CategorySets utilisée par un objet CalculationDefinition.
Propriété CharacterType	Définit ou retourne le type du jeu de caractères utilisé par une source de données.
Propriété ChildCategories	Retourne une collection Categories.
Propriété ChildCubes	Retourne une collection d'objets ChildCube à partir d'un groupe de cubes.
Propriété ChildCustomViews	Retourne une collection d'objets CustomView.
Propriété ChildMeasures	envoie une collection de mesures enfant qui proviennent du dossier de mesures.
Propriété Code	Définit ou retourne un code qui identifie de manière unique la catégorie dans toute la dimension.
Propriété Columns	Retourne une collection Columns pour une source de données.
Propriété ColumnsLoaded	Retourne une valeur indiquant si les colonnes d'une source de données ont été utilisées pour créer un modèle.
Propriété CompressMDC	Définit ou retourne une valeur indiquant si un cube est compressé pour économiser l'espace de stockage.
Propriété Connection	Définit et obtient la connexion à Content Manager.
Propriété Consolidate	Définit ou retourne une valeur indiquant le mode de consolidation d'un cube.
Propriété Context	Définit ou retourne la branche de niveaux utilisée pour classer des catégories au sein d'un niveau.
Propriété ContextLevel	Définit ou retourne une période de date.
Propriété ContextOffset	Définit ou retourne la position de la période de référence par rapport à la période en cours.
Propriété ConvergenceLevel	Retourne le niveau de convergence d'une branche de niveaux secondaire.
Propriété Count	Retourne le nombre d'objets dans une collection.

<b>Propriété</b>	<b>Description</b>
Propriété CountryCode	Définit ou retourne le code du pays ou de la région auquel s'applique un enregistrement de devise.
Propriété CubeCodePage	Définit ou retourne le paramètre de page de codes du cube du modèle utilisé pour créer le cube.
Propriété CubeCreation	Définit ou retourne une valeur indiquant si le cube est créé.
Propriété CubeCustomViews	Retourne une collection d'objets CustomView.
Propriété Cubes	Retourne une collection d'objets Cube et CubeGroup.
Propriété CubeStamp	Retourne la date et l'heure de création du cube.
Propriété CurrencyCountryLabel	Définit ou retourne le libellé du pays ou de la région de la devise.
Propriété CurrencyDecimals	Définit ou retourne le nombre de décimales utilisées dans une devise.
Propriété CurrencyFormatOverride	Définit ou retourne une valeur indiquant si vous pouvez écraser le format standard d'une devise.
Propriété CurrencyIsEMU	Définit ou retourne une valeur indiquant si l'enregistrement est un enregistrement de devise de l'UEM.
Propriété CurrencyIsEuro	Définit ou retourne une valeur indiquant si l'enregistrement est la devise euro de base.
Propriété CurrencyRates	Retourne une collection CurrencyRates.
Propriété CurrencyRecord	Retourne l'objet CurrencyRecord auquel s'applique l'objet CurrencyRate.
Propriété CurrencyRecords	Retourne une collection d'objets CurrencyRecord.
Propriété CurrencySymbol	Définit ou retourne le symbole monétaire associé à une devise.
Propriété CurrencyTable	Retourne un objet CurrencyTable associé à un taux de conversion.
Propriété CurrencyTables	Retourne une collection d'objets CurrencyTable.



<b>Propriété</b>	<b>Description</b>
Propriété CurrencyTableType	Définit ou retourne le type de la table des devises.
Propriété CurrentModel	Retourne l'objet Model actif.
Propriété CurrentValueIndex	Définit ou retourne l'index de valeur d'invite en cours.
Propriété CustomView	Définit ou retourne une valeur indiquant si une vue contient la totalité des catégories d'une dimension, une partie seulement ou aucune d'entre elles.
Propriété CustomViews	Retourne une collection d'objets CustomView.
Propriété DataCharacterSet	Définit ou retourne le jeu de caractères par défaut utilisé par l'application.
Propriété DataClass	Définit ou retourne le type de données d'une colonne source.
Propriété DataRange	Définit ou retourne le nom d'une plage de base de données dans une source de données.
Propriété DataSource	Définit ou retourne la source de données de Content Manager.
Propriété DataSourcePath	Définit ou retourne l'emplacement dans lequel Transformer recherche les fichiers de la source de données.
Propriété DataSources	Retourne une collection d'objets DataSource dans un modèle.
Propriété DataSourceWindowsLocation	Définit ou renvoie l'emplacement du cube, ce qui comprend le chemin d'accès complet et le nom du cube.
Propriété DataTemporaryFilePath	Définit ou retourne le nom du répertoire dans lequel Transformer crée des fichiers de travail temporaires lorsqu'il génère des cubes.
Propriété DateDegreeofDetail	Définit ou retourne le niveau de date auquel des rapports sont générés.
Propriété DateDegreeofDetailLevelName	Définit ou retourne le niveau de date qui s'applique à une mesure cumule extérieurement.

<b>Propriété</b>	<b>Description</b>
Propriété DateFormat	Définit ou retourne une valeur indiquant la manière dont les dates apparaissent.
Propriété DateFunction	Définit ou retourne les catégories de date qui sont générées dans un niveau.
Propriété DateInputFormat	Définit ou retourne l'ordre du format des dates dans la source de données.
Propriété DateLevel	Définit ou retourne le niveau d'une dimension de temps auquel s'appliquent les taux de conversion.
Propriété DateWizard	Retourne l'objet DateWizard.
Propriété DecimalPoint	Définit ou retourne le caractère utilisé comme virgule décimale.
Propriété Decimals	Retourne le nombre de décimales dans une colonne lorsqu'il est défini dans les données sources.
Propriété DefaultCategoryOrderBy	Définit l'ordre de tri par défaut pour toutes les catégories du modèle.
Propriété DefaultDateFormat	Définit ou retourne le paramètre par défaut de la propriété DateInputFormat.
Propriété Description	Définit ou retourne la description de l'objet.
Propriété DesiredPartitionSize	Définit ou retourne la taille de partition souhaitée.
Propriété DetachDataSource	Définit ou retourne une valeur indiquant si la connexion à la source de données est maintenue ou libérée.
Propriété DetailLevel	Définit ou retourne le niveau de détail le plus bas des cubes dans un objet CubeGroup.
Propriété Dimension	Retourne une dimension pour un objet Category.
Propriété DimensionInclude	Définit ou retourne le type de vue personnalisée.
Propriété DimensionLevels	Retourne une collection DimensionLevels.
Propriété DimensionName	Définit le nom d'un nouvel objet DateDimension.

Propriété	Description
Propriété Dimensions	Retourne une collection d'objets Dimension et DateDimension.
Propriété DimensionView	Définit ou retourne l'objet View associé à un cube ou à une vue personnalisée.
Propriété DimensionViewType	Définit ou retourne les dimensions et les vues qui appartiennent à un cube.
Propriété DisplayName	Définit ou renvoie le nom à afficher pour SecurityObject.
Propriété DrillCode	Définit ou retourne un code qui identifie de manière unique la catégorie de la branche de niveaux dans toute la dimension.
Propriété DrillDowns	Retourne une collection DrillDowns.
Propriété DrillInclusion	Définit ou retourne une valeur indiquant si une branche de niveaux est incluse dans un cube.
Propriété DrillThroughTargets	Retourne une collection d'objets cible d'accès au détail associés à un modèle Transformer.
Propriété DuplicateRollup	Définit ou retourne une valeur indiquant la manière dont les valeurs de mesure des enregistrements consolidés sont cumulées.
Propriété DuplicateWeight	Définit ou retourne le nom de la mesure qui contient des facteurs de pondération de moyenne.
Propriété EarliestDate	Définit la date la plus ancienne dans la plage de dates utilisée pour sélectionner des catégories.
Propriété EMUEntryDate	Définit ou retourne la date à laquelle commencent les calculs de triangulation de l'euro.
Propriété EnableMessageLogging	Définit ou retourne une valeur indiquant si les messages de Transformer sont consignés dans un fichier journal.
Propriété EnableTimePeriod	Définit le niveau de détail d'une dimension de temps.
Propriété EstimatedRows	Définit ou retourne une estimation du nombre d'enregistrements contenus dans le cube avant le découpage automatique en partitions.

Propriété	Description
Propriété ExcludeAutoPartition	Définit ou retourne une valeur indiquant si une dimension est exclue du processus de découpage automatique en partitions.
Propriété ExpressionText	Définit ou retourne le contenu d'une expression qui spécifie la valeur d'un objet.
Propriété External	Définit ou retourne une valeur indiquant si la source de données contient des valeurs récapitulatives.
Propriété FieldSeparator	Définit ou retourne le type de séparateur de zones utilisé par la source de données.
Propriété FileName	Renvoie le nom d'un fichier modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété Filters	Renvoie la collection d'objets Filter associés à une requête.
Propriété FindCategoryByCatCode	Retourne l'objet Category contenant la chaîne du code de catégorie spécifiée.
Propriété Format	Définit ou retourne une valeur indiquant la manière dont les valeurs numériques apparaissent.
Propriété FormatDecimals	Définit ou retourne le nombre de décimales que PowerPlay affiche pour la mesure.
Propriété FullName	Retourne l'emplacement d'un fichier modèle.
Propriété GenerateCategories	Définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.
Propriété GenerateDateCategories	Définit ou retourne une valeur indiquant si un niveau de date génère des catégories de date.
Propriété GenerateDates	Définit une valeur indiquant si l'objet DateWizard génère des catégories de date.
Propriété GeneratePowerCube	Définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.
Propriété GenerateTimePeriod	Définit ou retourne les options de génération de catégorie pour une dimension de temps.

Propriété	Description
Propriété Group	Définit ou retourne une valeur indiquant si un objet CalculationDefinition est regroupé avec les catégories de l'ensemble de catégories.
Propriété GroupDimension	Définit ou retourne la dimension utilisée pour créer le groupe de cubes.
Propriété GroupLevel	Définit ou retourne le niveau auquel appartiennent les catégories qui deviennent les cubes individuels d'un groupe de cubes.
Propriété HasSubdimension	Retourne une valeur indiquant si un objet de niveau contient une sous-dimension.
Propriété HideValue	Spécifie si la valeur d'un objet Category doit être masquée. Valeur par défaut : false.
Propriété ID	Définit ou retourne l'ID d'espace-noms.
Propriété IgnoreMissingValue	Spécifie si les valeurs nulles et manquantes doivent être ignorées dans un cumul d'état temporel. Valeur par défaut : false.
Propriété Inclusion	Définit ou retourne les circonstances dans lesquelles des catégories sont incluses dans un cube.
Propriété IncrementalUpdate	Définit ou retourne une valeur indiquant si la mise à jour incrémentielle d'un cube ou d'un groupe de cubes est appliquée à partir de la source de données.
Propriété InputScale	Définit ou retourne la valeur d'échelle utilisée pour convertir les nombres décimaux des colonnes en entiers.
Propriété IsAnyColumnMismatched	Retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.
Propriété IsBad	Retourne une valeur indiquant si le modèle en attente est corrompu ou récupérable.
Propriété IsExpressionValid	Définit ou retourne une valeur indiquant si une expression est valide.
Propriété IsFolder	Définit et renvoie une valeur qui indique si une mesure représente un dossier de mesures.
Propriété IsManual	Retourne une valeur indiquant si un niveau est associé à une valeur source.

<b>Propriété</b>	<b>Description</b>
Propriété IsMDCInUse	Retourne une valeur qui indique si un cube est en cours d'utilisation ou de construction.
Propriété IsolationLevel	Définit ou retourne le niveau d'isolement utilisé pour définir des transactions autorisées.
Propriété IsPrimary	Définit ou retourne une valeur indiquant si la catégorie de branche ou la branche de niveaux est la principale.
Propriété IsTimeBasedPartitionedCube	Définit ou retourne une valeur qui indique si un groupe de cubes est défini comme cube découpé en partitions de temps.
Propriété KeyName	Définit ou retourne la valeur qui apparaît dans la colonne associée de la source de données.
Propriété Label	Définit ou retourne un nom descriptif qui apparaît dans PowerPlay.
Propriété LastUseDate	Retourne la date de la dernière modification ou de la dernière utilisation de la catégorie.
Propriété LatestDate	Définit la date la plus récente dans la plage de dates utilisée pour sélectionner des catégories.
Propriété Level	Retourne un niveau pour un objet Category ou SpecialCategory.
Propriété LevelCategories	Définit ou retourne les catégories d'un niveau spécifique.
Propriété LevelDrillDowns	Retourne une collection LevelDrillDowns.
Propriété Levels	Retourne une collection Levels.
Propriété LocalPath	Définit ou retourne l'emplacement d'une source de données locale.
Propriété LogErrorLevel	Définit ou retourne le niveau de gravité des messages d'erreur consignés.
Propriété LogFileAppend	Définit ou retourne une valeur indiquant si Transformer ajoute des messages au fichier journal ou s'il écrase les messages antérieurs de journal.
Propriété LogFileName	Définit ou retourne le nom du fichier journal.

Propriété	Description
Propriété LogFilePath	Définit ou retourne l'emplacement dans lequel Transformer enregistre le fichier journal.
Propriété Lunar	Définit ou retourne une valeur indiquant si l'objet est basé sur une année lunaire.
Propriété ManualCurrentPeriod	Définit ou retourne une valeur indiquant si la période de date en cours est définie manuellement ou automatiquement.
Propriété MaximizeSpeed	Définit ou retourne une valeur indiquant si la vitesse est optimisée lors de la génération des catégories.
Propriété MaxNumPartLevels	Définit ou retourne le nombre maximum de fois que Transformer lit la source de données pendant le découpage en partitions d'un cube.
Propriété MaxTransactionNumber	Définit ou retourne le nombre maximum d'enregistrements que Transformer traite avant d'enregistrer les modifications dans un cube.
Propriété MDCFile	Définit ou retourne le nom d'un fichier de PowerCube (.mdc).
Propriété MeasureInclude	Définit ou retourne le nom d'une mesure à inclure dans un cube, dans un groupe de cubes ou dans une classe d'utilisateurs.
Propriété MeasureName	Définit ou retourne un titre descriptif qui identifie une mesure sur la ligne des dimensions de PowerPlay.
Propriété Measures	Retourne une collection Measures.
Propriété MeasureType	Retourne une valeur indiquant s'il s'agit d'une mesure ordinaire, d'une mesure calculée ou d'une mesure de nombre de catégories.
Propriété MissingValue	Définit ou retourne une valeur indiquant ce qui est affiché à la place d'une valeur vide ou nulle.
Propriété ModelName	Retourne le nom d'un modèle en attente.
Propriété ModelsPath	Définit ou retourne l'emplacement dans lequel Transformer ouvre et enregistre les fichiers modèles.

<b>Propriété</b>	<b>Description</b>
Propriété ModelTemporaryFilePath	Définit ou renvoie l'emplacement dans lequel Transformer crée des fichiers modèles temporaires (.qy?).
Propriété ModelType	Renvoie l'extension d'un fichier modèle tel qu'elle apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété MonthType	Définit le mode de calcul du niveau mois d'une dimension de temps.
Propriété Name	Définit ou retourne le nom d'un objet.
Propriété Namespaces	Renvoie une collection Namespaces.
Propriété NewCatsLocked	Définit ou renvoie une valeur indiquant si vous pouvez ajouter de nouvelles catégories.
Propriété ObjectCAMID	Renvoie le CAMID de l'objet dans l'ensemble d'espaces-noms par la propriété ObjectName.
Propriété ObjectName	Définit ou retourne le nom d'un objet NameSpace.
Propriété Optimize	Définit ou retourne l'option d'optimisation du cube en cours.
Propriété OrderByDescending	Définit ou retourne une valeur indiquant si les valeurs apparaissent dans l'ordre croissant.
Propriété OrderByStorageType	Définit ou retourne une valeur indiquant la manière dont les catégories sont triées en fonction du type de stockage d'une colonne.
Propriété Origin	Retourne l'origine de l'objet spécifié.
Propriété OriginalName	Définit ou retourne le nom de la colonne dans la source de données.
Propriété Orphanage	Définit ou retourne une valeur indiquant si une catégorie est isolée.
Propriété OutputScale	Définit ou retourne la valeur d'échelle utilisée pour convertir les nombres entiers en nombres décimaux dans PowerPlay.
Propriété Packages	Renvoie la collection des objets Package associés à un modèle Transformer.



Propriété	Description
Propriété PackagesDatasourceConnections	Renvoie la collection d'objets PackageDatasourceConnection associée à un objet Package ou Report.
Propriété Parent	Retourne le nom du parent d'un objet.
Propriété ParentCategories	Retourne une collection de catégories parents pour une catégorie.
Propriété Partition	Définit ou retourne le numéro d'une partition manuelle.
Propriété Password	Définit un mot de passe qui fait la distinction entre les majuscules et les minuscules.
Propriété PatFile	Définit ou retourne l'emplacement du fichier de modèle cogtr_ <i>langue</i> .pat, tel que cogtr_en.pat, pour l'environnement local du produit associé.
Propriété Path	Retourne l'emplacement d'un fichier modèle.
Propriété PopulateByDataSource	Définit ou retourne une valeur indiquant si le taux de conversion est obtenu à partir d'une source de données ou s'il est indiqué dans Transformer.
Propriété Position	Définit ou retourne le nombre ordinal ou la position de départ d'une colonne dans la source de données.
Propriété PowerCubesPath	Définit ou retourne l'emplacement dans lequel Transformer crée des fichiers de PowerCube (.mdc).
Propriété PowerPlayPath	Définit ou retourne l'emplacement du fichier exécutable PowerPlay.exe.
Propriété Precision	Définit ou retourne le nombre de décimales des mesures utilisées dans des calculs.
Propriété PromptForPassword	Indique si les utilisateurs sont toujours invités à saisir un mot de passe lorsqu'ils utilisent Transformer en mode IU. S'applique uniquement au type de code d'accès DataSource.
Propriété Prompts	Renvoie la collection d'objets Prompt associés à une requête.
Propriété PromptValueType	Définit ou retourne le type d'invite.

Propriété	Description
Propriété QualifiedName	Retourne le nom complet d'un niveau.
Propriété QuarterType	Définit le mode de calcul du niveau trimestre d'une dimension de temps.
Propriété Queries	Renvoie la collection d'objets Query associée à un objet Package ou Report.
Propriété QyPath	Retourne l'emplacement d'un modèle en attente.
Propriété Rate	Définit ou retourne le taux de change d'une devise.
Propriété RefName	Définit ou retourne le nom du filtre dans la source de données.
Propriété RefreshDescription	Définit ou retourne une valeur indiquant si les descriptions sont mises à jour.
Propriété RefreshLabel	Définit ou retourne une valeur indiquant si les libellés sont mis à jour.
Propriété RefreshShortName	Définit ou retourne une valeur indiquant si les noms abrégés sont mis à jour.
Propriété RegularRollup	Définit ou retourne la fonction de cumul ordinaire en cours d'un objet.
Propriété RegularWeight	Définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.
Propriété Reports	Renvoie la collection des objets Report associés à un modèle Transformer.
Propriété ReverseSign	Définit ou retourne une valeur indiquant si PowerPlay inverse le signe d'une mesure.
Propriété Rollup	Définit ou retourne une valeur indiquant si les valeurs de mesure d'une catégorie spéciale sont cumulées dans la catégorie parent.
Propriété RollupTiming	Définit ou retourne une valeur indiquant à quel moment les calculs des mesures calculées doivent être effectués.
Propriété RowsAsSample	Définit ou retourne le nombre de lignes que la méthode DoAutoDesign utilise comme échantillon pour créer un modèle.

<b>Propriété</b>	<b>Description</b>
Propriété RowsChecked	Définit ou retourne le nombre maximum de lignes que la méthode DoAutoDesign lit à partir de la source de données.
Propriété RunningPeriods	Définit ou retourne le nombre de périodes de date utilisées dans les sommes cumulatives.
Propriété SecurityObjects	Renvoie une collection de SecurityObjects.
Propriété Server	Définit ou retourne une valeur indiquant si un cube est traité localement ou sur un serveur.
Propriété ServerModelPath	Définit ou retourne l'emplacement du serveur d'un modèle.
Propriété ServerPath	Définit ou retourne l'emplacement du serveur d'une source de données.
Propriété ServerQuery	Définit ou retourne une valeur indiquant si les données sont traitées localement ou sur un serveur.
Propriété ServicesBuildNumber	Retourne le numéro de version de Transformer au format numérique.
Propriété ServicesVersionText	Retourne le numéro de version de Transformer au format texte.
Propriété SetsCurrentPeriod	Définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.
Propriété ShortName	Définit ou retourne le nom abrégé de la mesure.
Propriété Signon	Définit ou retourne l'objet de code d'accès utilisé par un cube ou un groupe de cubes.
Propriété SignonNamespace	Contient l'espace-noms sécurisé associé au code d'accès IBM Cognos.
Propriété Signons	Retourne la collection Signons d'un modèle.
Propriété SignonType	Définit ou retourne le type de code d'accès. Le type de code d'accès peut être Datasource ou IBM Cognos.
Propriété Size	Définit ou retourne la taille d'une colonne ou d'un fichier modèle (.mdl).

Propriété	Description
Propriété SortComparisonRule	Définit ou retourne la règle de comparaison utilisée par Transformer pour trier des données.
Propriété SourceType	Définit ou retourne le type de fichier de données utilisé par une source de données.
Propriété SpecialCategoryCount	Retourne le nombre de changements de niveaux, de racines et de catégories spéciales dans une dimension.
Propriété SQLExpression	Retourne l'expression SQL utilisée pour définir un fichier de définition de requête d'Impromptu (.iqd).
Propriété Status	Retourne le statut d'un problème associé au cube lors de la dernière création.
Propriété StorageType	Définit ou retourne la taille d'un type de données numérique.
Propriété StreamExtractAllowed	Définit ou renvoie un indicateur qui détermine si les extractions de flux sont autorisées (s'applique uniquement à la source de données SAP BW).
Propriété StreamExtractSize	Définit ou renvoie la taille, en mégaoctets, du tampon utilisé pour transférer des données depuis SAP quand StreamExtract est défini sur True.
Propriété SummaryLevel	Définit ou retourne le niveau à utiliser pour récapituler des catégories externes dans un groupe de cubes.
Propriété SuppressNull	Définit ou renvoie l'option de suppression des valeurs nulles utilisée pour les sources de données SAP BW.
Propriété SuspendedModels	Retourne une collection d'objets SuspendedModel.
Propriété TargetLevel	Définit ou retourne le niveau de détail d'une période de date.
Propriété TargetOffset	Définit ou retourne la position de la période cible par rapport à la période en cours.
Propriété ThousandPoint	Définit ou retourne le caractère qui sépare les milliers.

Propriété	Description
Propriété Time	Renvoie l'horodateur d'un modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.
Propriété TimeArrayColumn	Définit ou retourne le nom de la première colonne du tableau quand l'objet représente un tableau de dates.
Propriété TimeArrayStartMonth	Définit ou retourne le mois auquel commence l'exercice fiscal quand l'objet inclut un tableau de dates.
Propriété TimeArrayType	Définit ou retourne le type de tableau utilisé pour les valeurs de date.
Propriété TimeRank	Définit ou retourne le rang relatif des niveaux de date dans une dimension de temps.
Propriété TimeStamp	Définit ou renvoie l'horodatage d'un objet Package ou Report.
Propriété TimeStateRollup	Définit ou retourne la période de date utilisée pour les cumuls d'état temporel.
Propriété TimeStateWeight	Définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.
Propriété ToDateLevel	Définit ou retourne la période de date utilisée pour les totaux jusqu'à ce jour.
Propriété TransdaPath	Définit ou retourne l'emplacement du fichier exécutable transda.exe.
Propriété TransformerSignon	Définit ou renvoie l'objet de code d'accès Transformer, associé à une connexion de source de données de pack (code d'accès IBM Cognos).
Propriété Type	Définit ou retourne le type d'un objet.
Propriété Unique	Définit ou retourne une valeur indiquant si Transformer peut identifier chaque catégorie du niveau à l'aide d'une valeur source unique.
Propriété UniqueMove	Définit ou retourne une valeur indiquant la manière dont un niveau unique est traité en cas de déplacement de la catégorie associée.

<b>Propriété</b>	<b>Description</b>
Propriété UseAltMDCFile	Définit ou retourne une valeur qui indique si un nom de fichier temporaire peut être utilisé.
Propriété User	Définit ou retourne un nom d'utilisateur associé à l'espace- noms.
Propriété UserCAMID	Renvoie le CAMID de l'utilisateur défini par la propriété User.
Propriété UserID	Définit ou retourne l'ID utilisateur du code d'accès.
Propriété Value	Définit ou retourne la valeur d'invite.
Propriété ValuesCount	Renvoie le nombre d'ensembles de valeurs pour l'invite.
Propriété Version	Retourne le numéro de version de Transformer.
Propriété Views	Retourne une collection Views.
Propriété ViewType	Définit ou retourne une valeur indiquant si une vue contient la totalité des catégories d'une dimension, une partie seulement ou aucune d'entre elles.
Propriété WeekAdd	Définit ou retourne le nombre de jours qui sont ajoutés à une année lunaire.
Propriété WeekSpan	Définit ou retourne une valeur indiquant la manière dont il faut traiter une semaine qui s'étale sur deux années.
Propriété WeekStart	Définit ou retourne le premier jour de la semaine.
Propriété WeekStartDay	Définit le premier jour de la semaine.
Propriété WorkingDay	Définit ou retourne une valeur indiquant si un jour spécifique fait partie de la semaine de travail.
Propriété WorkingDays	Définit ou retourne une valeur indiquant les jours qui font partie de la semaine de travail.
Propriété YearStartDay	Définit ou retourne le premier jour d'une année.
Propriété YearType	Définit le mode de calcul du niveau année d'une dimension de temps.

---

## Propriété ActivityMeasure

La propriété ActivityMeasure définit ou retourne la mesure utilisée par une mesure de nombre de catégories.

### Syntaxe

*Measure* .ActivityMeasure

### Application

Objet Measure

### Explication

Il est possible de définir une mesure dénombrant les catégories. Par exemple, vous pouvez créer une mesure qui indique le nombre de clients différents ayant acheté un produit donné chaque mois, chaque trimestre ou chaque année. Basé sur un niveau unique (notamment le numéro de client), ce nombre inclut toutes les valeurs non manquantes et non nulles, mais pas les doubles comptages. Si un même client achète un produit à deux reprises au cours du même trimestre, par exemple, le cumul trimestriel ne comptabilise ce client qu'une seule fois.

Utilisez la propriété ActivityMeasure pour spécifier une mesure d'activité particulière pour un nombre de catégories. Si vous ne définissez pas cette propriété, les résultats sont générés sur base de toutes les mesures du modèle qui répondent aux critères.

Une mesure d'activité ne peut pas être une mesure calculée, une mesure après cumul ou une mesure cumulée extérieurement.

Pour créer une mesure de nombre de catégories, ajoutez une mesure à la collection Measures et affectez à la propriété CategoryCountLevel le niveau auquel les catégories doivent être comptabilisées. Le niveau spécifié doit être unique.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet - Measure

### Accès

Lecture/écriture

### Exemples

```
objMeasure.CategoryCountLevel = objLevelobjMeasure.ActivityMeasure  
= objActMeasure
```

---

## Propriété Aggregate

La propriété Aggregate définit ou retourne le type de catégorie de date relative.

## Syntaxe

*SpecialCategory* .Aggregate

## Application

Objet SpecialCategory

## Explication

Cette propriété permet de personnaliser une catégorie que vous pouvez ensuite utiliser pour effectuer un suivi des mesures pour des périodes de date spécifiques relatives à la date en cours. Les catégories de date relative peuvent spécifier :

- une période unique, pour illustrer les changements d'une période de date à une autre,
- un ensemble séquentiel de périodes, pour illustrer des totaux à ce jour qui comparent les performances actuelles aux performances passées,
- une période de date relative pour illustrer des totaux cumulatifs, notamment une période de six mois avant la date en cours.

La propriété Aggregate utilise les valeurs de xtrTimeAggregate.

Les propriétés de l'exemple ci-dessous spécifient une catégorie de total cumulatif qui s'étend sur les six mois qui précèdent le mois en cours, pour cette année et l'année dernière.

```
Aggregate = trAggregateRunningGrouped
ContextLevel = "Year"
ContextOffset = -1
RunningPeriods = 6
TargetLevel = "Month"
TargetOffset = 0
```

## Type

Constante - xtrTimeAggregate

## Accès

Lecture/écriture

## Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories.
  Add(xtrObjectType.trSpecialCategory)objSpecCategory.Aggregate
= xtrTimeAggregate.trAggregateRunning
```

---

## Propriété AllocationMeasure

La propriété AllocationMeasure retourne l'objet Measure utilisé comme facteur de pondération.

## Syntaxe

*object* .AllocationMeasure(Measure)



## Application

Objet Category

Objet Dimension

Objet DateLevel

Objet DateDimension

Objet Level

Objet SpecialCategory

## Explication

Une mesure qui est allouée de manière proportionnelle à des catégories descendantes requiert l'emploi d'une deuxième mesure comme facteur de pondération. Utilisez cette propriété avec la propriété AllocationType pour déterminer les paramètres d'allocation en cours.

Par exemple, vous pouvez allouer la valeur d'une mesure de coûts fixes à diverses régions en fonction d'une autre mesure, telle que les ventes dans chaque région.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requise. Spécifie la mesure qui est allouée de manière proportionnelle aux catégories descendantes.  Type : Objet

## Type

Objet - Measure

## Accès

Lecture

---

## Propriété AllocationType

La propriété AllocationType renvoie le mode d'allocation d'une mesure par un objet.

## Syntaxe

*object* .AllocationType(Measure)

## Application

Objet Category

Objet Dimension

Objet DateLevel

Objet DateDimension

Objet de niveau

Objet SpecialCategory

## Explication

Vous pouvez allouer une mesure à des catégories descendantes de manière proportionnelle ou sous forme de constante, voire même ne pas l'allouer. Utilisez cette propriété avec la propriété AllocationMeasure pour déterminer les paramètres d'allocation en cours.

La propriété Allocation utilise les valeurs de la constante xtrAllocationType.

Paramètre	Description
Mesure	Requise. Spécifie la mesure d'allocation à supprimer ou à utiliser avec des catégories descendantes.  Type : Objet - Mesure

## Type

Constante - xtrAllocationType

## Accès

Lecture

## Exemples

```
objLevel  
= objModel.Dimensions("Products").DimensionLevels(1)If  
objLevel.AllocationType(objMeasure)  
<> _xtrAllocationType.trAllocationByAnotherMeasure Then
```

---

## Propriété AllowCurrencyConversion

La propriété AllowCurrencyConversion définit ou retourne une valeur indiquant si une mesure qui représente une devise peut être convertie en une autre devise dans PowerPlay.

## Syntaxe

*Measure* .**AllowCurrencyConversion**

## Application

Objet Measure

## Explication

Définissez cette propriété uniquement après avoir défini une table des devises. Les devises susceptibles d'être sélectionnées par un utilisateur de PowerPlay sont limitées à celles définies dans la collection CurrencyRecords.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objMeasure.CategoryCountLevel = objLevel.AllowCurrencyConversion  
= False
```

---

## Propriété AllowDrillThrough

La propriété AllowDrillThrough définit ou retourne une valeur indiquant si un cube ou une mesure peut accéder au détail d'un cube ou d'un rapport.

## Syntaxe

*object* .AllowDrillThrough

## Application

Objet ChildCube

Objet Cube

Objet CubeGroup

Objet Measure

## Explication

Cette propriété permet de déterminer si un cube ou une mesure peut accéder au détail d'un fichier externe tel qu'un rapport Impromptu ou un fichier de PowerCube.

Utilisez d'abord un objet Report pour créer une liaison d'accès au détail, puis affectez la valeur True à la propriété AllowDrillThrough pour activer la fonction d'accès au détail.

Valeur par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objMeasure = objModel.Measures.Item("Quantity")objMeasure.AllowDrillThrough  
= True
```

---

## Propriété AlternateQueryPath

La propriété AlternateQueryPath définit ou retourne le nom d'une autre source de données pour un cube.

## Syntaxe

```
object .AlternateQueryPath(DataSource)
```

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété avec le paramètre DataSource pour commander à Transformer de lire les données à partir d'autres sources de données quand vous créez un cube. Vous devez d'abord utiliser les outils d'accès fournis avec votre logiciel de base de données pour configurer des fichiers sources de remplacement dont la structure est identique à celle des fichiers sources sur lesquels le modèle est basé. La source de données de remplacement doit posséder les mêmes colonnes que la source d'origine pour être lue dans le modèle Transformer.

Vous décidez, par exemple, de créer des cubes pour les différentes divisions de ventes à partir de fichiers sources différents, mais dans tous les cas vous pouvez utiliser le même modèle d'analyse des ventes. Définissez plusieurs fichiers de définition de requête d'Impromptu (.iqd), contenant chacun des filtres permettant d'extraire les données pour une seule division des ventes.

Paramètre	Description
DataSource	Requis pour définir la propriété. Spécifie l'objet utilisé pour référencer la source de données de remplacement.  Type : Objet

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété AltMDCFile

La propriété AltMDCFile spécifie un nom de fichier secondaire pour le cube.

### Syntaxe

*object* .AltMDCFile

### Application

Objet Cube

Objet CubeGroup

Objet ChildCube

### Explication

Utilisez cette propriété pour indiquer un nom de fichier qui sera utilisé quand le cube en cours d'utilisation dans une autre application. Afin d'activer cette propriété, la valeur de la propriété AltMDCFile doit être définie à 'true'.

La propriété IsMDCInUse permet de déterminer si un cube est en cours d'utilisation.

Cette propriété est facultative. Si la valeur de la propriété UseAltMDCFile est définie à true, le nom de fichier secondaire doit être indiqué à l'aide de cette propriété. Si aucune valeur n'est définie pour cette propriété, Transformer utilise d'office un nom secondaire par défaut. Transformer ajoute un chiffre à la fin du nom de fichier, en commençant par un et ainsi de suite, jusqu'à ce que le nom de fichier soit accepté.

### Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété AlwaysUseTransformerSignon

La propriété AlwaysUseTransformerSignon indique s'il faut utiliser le code d'accès Transformer ou Content Manager.

### Syntaxe

*PackageDataSourceConnection* .AlwaysUseTransformerSignon

## Application

Objet PackageDatasourceConnection

## Explication

Si la valeur est True, le code d'accès Transformer est prioritaire sur le code d'accès Content Manager. Lorsque la valeur est False, le code d'accès de Content Manager est utilisé par défaut.

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
package = model.Packages.Add()connection.AlwaysUseTransformerSignon  
= True
```

---

## Propriété Apex

La propriété Apex définit ou retourne l'objet Category utilisé comme racine de la vue d'une dimension.

## Syntaxe

*View* .Apex

## Application

Objet View

## Explication

Cette propriété permet de limiter les catégories qu'un utilisateur peut afficher dans PowerPlay au seul sous-ensemble dont il a besoin.

Vous pouvez créer une vue des dimensions et appliquer la propriété Apex à la catégorie spécifiée. Une fois que vous avez créé un cube, il ne contient que la catégorie du sommet et ses descendants directs. Les ancêtres, les soeurs et les descendants de celles-ci sont tous omis de la vue.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objProductsDim = objModel.Dimensions.Item("Products")objViewItem.Apex  
= objProductsDim.Drilldowns(1).Categories(1)
```

---

## Propriété Application

La propriété Application retourne l'objet Transformer Application.

### Syntaxe

*objet* .Application

### Application

Collection Associations

Objet Association

Collection CalculationDefinitions

Objet CalculationDefinition

Collection Categories

Objet Category

Collection CategorySets

Objet CategorySet

Collection Columns

Objet Column

Objet CrossTabDataSource

Collection Cubes

Objet Cube

Collection CubeCustomViews

Objet CubeGroup

Collection ChildCubes

Objet ChildCube

Collection CurrencyRecords

Objet CurrencyRecord

Collection CurrencyRates

Objet CurrencyRate

Collection CurrencyTables  
Objet CurrencyTable  
Collection CustomViews  
Objet CustomView  
Objet DateDimension  
Objet DateLevel  
Collection DataSources  
Objet DataSource  
Objet DateWizard  
Objet DbDataSource  
Objet DateDrillDown  
Collection Dimensions  
Objet Dimension  
Collection DimensionLevels  
Collection DrillDowns  
Objet DrillDown  
Collection DrillThroughTargets  
Objet DrillThroughTarget  
Objet FlatFileDataSource  
Collection Filters  
Objet Filter  
Objet IqdDataSource  
Collection PackageDatasourceConnections  
Objet PackageDatasourceConnection  
Collection Packages  
Objet Package  
Collection Prompts  
Objet Prompt



Collection Queries

Objet Query

Collection Levels

Objet de niveau

Collection LevelDrillDowns

Collection Measures

Objet Measure

Objet Model

Collection Namespaces

Objet Namespace

Collection Names

Objet Name

Collection Reports

Objet Report

Collection SecurityObjects

Objet SecurityObject

Collection SuspendedModels

Objet SuspendedModel

Collection Signons

Objet de code d'accès

Objet SpecialCategory

Collection Views

Objet View

## **Explication**

Cette propriété permet de référencer des propriétés de l'objet Application Transformer à partir d'autres objets et d'autres collections du modèle.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture

---

## Propriété AssociationRole

La propriété AssociationRole définit ou retourne le rôle d'un objet Association.

## Syntaxe

*Association* .AssociationRole

## Application

Collection Associations

## Explication

Les tables des devises, les dimensions de date, les niveaux, les niveaux de date, les dimensions et les mesures entretiennent une relation avec les données de leur source sous-jacente par l'intermédiaire d'une collection Associations. Chaque objet Association de la collection utilise la propriété AssociationRole pour déterminer le rôle.

Les catégories d'un niveau, par exemple, peuvent obtenir leurs valeurs sources d'une colonne, telle que le code du produit, et leurs valeurs de libellé d'une autre colonne, notamment le nom du produit. Dans ce cas, la collection contient deux objets Association ayant chacun un rôle différent.

La propriété AssociationRole utilise les valeurs de la constante xtrAssociationRole.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - Associations

## Accès

Lecture/écriture

## Exemples

```
objAssociation  
= objDrill.ConvergenceLevel.Associations.Add()objAssociation.AssociationRole  
= xtrAssociationRole.trAssociationOrderBy
```

---

## Propriété Associations

La propriété Associations retourne une collection d'objets Association.

## Syntaxe

*object* .Associations

## Application

Objet CurrencyTable

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet Measure

## Explication

Plusieurs types d'objets dans un modèle entretiennent une relation avec la colonne de leur source sous-jacente par l'intermédiaire d'une collection Associations. L'association définit la source des valeurs de l'objet.

Les catégories d'un niveau, par exemple, peuvent obtenir leurs valeurs sources d'une colonne telle que le code du produit, et leurs valeurs de libellé à partir d'une autre colonne comme le nom du produit. Dans ce cas, la collection contient deux objets Association ayant chacun un rôle différent.

Pour les objets Measure, l'association n'est valide que si la propriété MeasureType de l'objet a pour valeur la constante trRegularMeasure. Une mesure qui dérive ses valeurs d'une association ne peut pas utiliser les propriétés ExpressionText et CategoryCountLevel pour dériver des valeurs, car elles s'excluent mutuellement.

## Type

Objet

## Accès

Lecture

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()
```

---

## Propriété AssociationType

La propriété AssociationType définit ou retourne une valeur indiquant si un objet Association possède une association avec une source de données.

## Syntaxe

*Association* .AssociationType

## Application

Objet Association

## Explication

La propriété AssociationType utilise les valeurs de la constante xtrAssociationType.

## Type

Constante - xtrAssociationType

## Accès

Lecture/écriture

## Exemples

```
objAssociation  
= objDrill.ConvergenceLevel.Associations.Add()objAssociation.AssociationType  
= xtrAssociationType.trAssociationQuery
```

---

## Propriété AutoLogon

Si la valeur de la propriété AutoLogon est True, elle indique si l'application s'authentifie automatiquement auprès des espaces-noms sécurisés associés au code d'accès.

## Syntaxe

*Query* .AutoSummary

## Application

Objet de code d'accès

## Explication

Si cette propriété est définie sur True, l'application s'authentifie automatiquement auprès des espaces-noms sécurisés associés au code d'accès.

La propriété AutoLogon s'applique uniquement au type CognosSignon.

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
signon = model.Signons.Add()signon.AutoLogon = True
```

---

## Propriété AutoSummary

La propriété AutoSummary définit ou retourne l'option AutoSummary pour une requête.

### Syntaxe

*query* .AutoSummary

### Application

Objet Query

### Explication

Cette propriété doit être définie pour un pack. Elle ne doit pas être définie pour un rapport.

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
new_package = model.Packages.Add()new_query.AutoSummary  
= True
```

---

## Propriété BlankSubstitute

La propriété BlankSubstitute définit ou retourne le libellé par défaut des catégories en blanc générées dans un niveau.

### Syntaxe

*object* .BlankSubstitute

### Application

Objet DateLevel

Objet de niveau

### Explication

Si aucune valeur de la source n'est associée à une catégorie ordinaire, la chaîne spécifiée par cette propriété est affectée comme libellé par défaut. De cette façon, vous pouvez inclure intentionnellement des catégories possédant des valeurs en blanc dans une source de données afin de préserver une hiérarchie particulière des catégories.

### Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")objLevel  
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill  
= objLevel.CreateAlternateDrillDownobjNewLevel  
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.BlankSubstitute  
= "No Value"
```

---

## Propriété BlockParentTotals

La propriété BlockParentTotals définit ou retourne une valeur indiquant si des parents d'enfants exclus affichent une valeur refusée ou le total des enfants non exclus.

## Syntaxe

*Cube* .BlockParentTotals

## Application

Objet Cube

## Explication

Utilisez cette propriété facultative pour garantir que les parents d'enfants exclus affichent une valeur refusée et non le total des enfants non exclus. Ceci empêche l'affichage de données représentant le cumul inexact des catégories non exclues uniquement. Les valeurs manquantes prévalent sur celles refusées et continuent d'afficher la valeur zéro, non disponible ou en blanc (cellule vide), voire les valeurs manquantes, selon la façon dont les mesures dans le cube ont été conçues pour traiter les valeurs manquantes. Ceci permet à l'utilisateur de distinguer les valeurs manquantes des valeurs refusées.

**Paramètre par défaut :** False

**Remarque :** Les cubes créés dans la version 6.6 ou antérieure de Transformer continuent de se comporter comme décrit précédemment quelle que soit la version dans laquelle ils sont ouverts sauf si l'utilisateur assigne la valeur TRUE à la propriété BlockParentTotals une fois le cube importé dans la version 7.0.

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.BlockParentTotals  
= True
```

---

## Propriété CacheCrossTabs

La propriété CacheCrossTabs définit ou retourne une valeur indiquant si des récapitulatifs sont mises en cache.

### Syntaxe

*object* .CacheCrossTabs

### Application

Objet Cube

Objet CubeGroup

### Explication

Cette propriété permet de stocker des récapitulatifs dans un cube pour le tableau croisé PowerPlay initial. (Le tableau initial affiche la première dimension sous forme de lignes et la deuxième sous forme de colonnes.) Cela contribue à optimiser le temps d'accès dans PowerPlay pour ce tableau initial seulement.

Paramètre par défaut : False

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objCube = objModel.Cubes.Item(1)objCube.CacheCrossTabs  
= True
```

---

## Propriété CalculationDefinitions

La propriété CalculationDefinitions retourne une collection des objets CalculationDefinition associés à une dimension.

### Syntaxe

*object* .CalculationDefinitions

### Application

Objet DateDimension

Objet Dimension

### Explication

Les objets CalculationDefinition sont constitués d'expressions complexes qui incluent dans le calcul les valeurs d'une ou plusieurs catégories.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Objet - CalculationDefinitions

### **Accès**

Lecture

### **Exemples**

```
calcDef = dimension.CalculationDefinitions.Add()
```

---

## **Propriété CAMID**

La propriété CAMID renvoie le CAMID de l'espace-noms.

### **Syntaxe**

*Namespace* .CAMID

### **Application**

Objet Namespace

### **Explication**

Cette propriété permet d'extraire le CAMID d'un espace-noms.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Chaîne

### **Accès**

Lecture

### **Exemples**

```
CAMID_of_Namespace = new_namespace.CAMID
```

---

## **Propriété CanAllocate**

La propriété CanAllocate retourne une valeur indiquant si vous pouvez allouer des valeurs de mesure aux niveaux descendants et aux catégories.

### **Syntaxe**

*object* .CanAllocate



## Application

Objet Category

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet SpecialCategory

## Explication

Cette propriété permet de déterminer si vous pouvez allouer les mesures d'une source de données à des catégories et à des niveaux associés à une autre source de données. La fonction d'allocation distribue des données, définies au niveau récapitulatif d'une dimension, à des niveaux inférieurs. Vous pouvez allouer sur :

- une dimension entière, lorsque la mesure apparaît dans une source de données ne désignant pas la dimension,
- les niveaux d'une dimension, lorsque la mesure est déjà spécifiée à un niveau de cette dimension,
- les catégories d'un niveau, lorsque la mesure est spécifiée à ce niveau particulier.

Par exemple, les recettes des ventes peuvent être suivies quotidiennement, tandis que les prévisions peuvent être établies pour chaque trimestre. L'allocation par proportion peut s'avérer utile pour répartir les prévisions trimestrielles aux niveaux du mois et du jour.

Cette propriété ne prend la valeur True que lorsque le modèle utilise au moins deux sources de données.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Booléen

## Accès

Lecture

## Exemples

```
objLevel = objModel.Dimensions("Products").DimensionLevels(1) If objLevel.CanAllocate = True Then
```

---

## Propriété CanAllocateByMeasure

La propriété CanAllocateByMeasure retourne une valeur indiquant si vous pouvez utiliser la mesure spécifiée comme facteur de pondération quand vous effectuez l'allocation par mesure.

## Syntaxe

*object* .CanAllocateByMeasure(Measure)

## Application

Objet Category

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet SpecialCategory

## Explication

Quand une mesure est allouée de manière proportionnelle, elle utilise les valeurs d'une autre mesure comme facteur de pondération. Les utilisateurs de PowerPlay voient la valeur s'afficher dans toutes les catégories descendantes. Cette propriété permet de déterminer si la mesure spécifiée peut être utilisée comme facteur de pondération.

Utilisez la méthode SetAllocation pour allouer les valeurs d'une mesure à toutes les catégories descendantes. La fonction d'allocation distribue des données, définies au niveau récapitulatif d'une dimension, à des niveaux inférieurs.

Par exemple, vous pouvez allouer la valeur d'une mesure de coûts fixes à diverses régions en fonction d'une autre mesure, telle que les ventes dans chaque région.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requis. Spécifie l'objet Measure utilisé comme facteur de pondération.  Type : Objet - Measure

## Type

Booléen

## Accès

Lecture

## Exemples

```
objLevel  
= objModel.Dimensions("Products").DimensionLevels(1)If  
  objLevel.CanAllocateByMeasure(objByMeasure)  
= True Then
```

---

## Propriété CanAllocateMeasure

La propriété CanAllocateMeasure retourne une valeur indiquant si vous pouvez allouer la mesure spécifiée aux niveaux descendants et aux catégories.

### Syntaxe

*object* .CanAllocateMeasure(Measure)

### Application

Objet Category

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

Objet SpecialCategory

### Explication

Cette propriété permet de déterminer si une mesure spécifique peut être utilisée pour l'allocation. Si vous souhaitez allouer la mesure de manière proportionnelle, vous pouvez utiliser la propriété CanAllocateByMeasure pour déterminer les mesures susceptibles d'être employées comme facteur de pondération.

Utilisez la méthode SetAllocation pour allouer les valeurs d'une mesure à toutes les catégories descendantes. La fonction d'allocation distribue des données, définies au niveau récapitulatif d'une dimension, à des niveaux inférieurs.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Mesure	Requis. Spécifie l'objet Measure utilisé comme facteur de pondération.  Type : Objet

### Type

Booléen

## Accès

Lecture

## Exemples

```
objLevel  
= objModel.Dimensions("Products").DimensionLevels(1)If  
objLevel.CanAllocateMeasure(objMeasure)  
= True Then
```

---

## Propriété Categories

### Description

La propriété Categories renvoie une collection Categories associée à un objet.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Syntaxe

*object* .Categories

### Application

Objet CategorySet

Objet DateDimension

Objet Dimension

Objet DateDrillDown

Objet DrillDown

### Explication

Dans le cas d'un objet CategorySet, cette propriété permet de retourner les catégories utilisées dans un calcul.

Avec un objet Dimension, elle retourne la collection des catégories spéciales. Elle peut aussi retourner les catégories calculées qui sont connectées à des catégories spéciales.

S'il s'agit d'un objet DrillDown, cette propriété retourne toutes les catégories utilisées dans un niveau. Utilisez la propriété ChildCategories pour retourner une collection de catégories enfants.

### Type

Objet

## Accès

Lecture

## Exemples

```
objCategory = objModel.Dimensions("Retailers").DrillDowns(1).Categories(2)
```

---

## Propriété Category

La propriété Category retourne la catégorie de date applicable pour un taux de conversion donné.

### Syntaxe

*CurrencyRate* .Category

### Application

Objet CurrencyRate

### Explication

Les taux de conversion doivent être liés à une catégorie de date (jour, semaine, mois, trimestre ou année) pour être valides. Si des taux sont définis pour chaque mois, par exemple, cette propriété retourne le mois de l'année associé à un taux de conversion spécifique.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet

### Accès

Lecture

### Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRate  
= objCurrencyRecord.CurrencyRates(intX)Select Case objCurrencyRate.Category.KeyName
```

---

## Propriété CategoryCount

La propriété CategoryCount retourne le nombre de catégories ordinaires dans une dimension ou un niveau.

### Syntaxe

*object* .CategoryCount

### Application

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

### Explication

Cette propriété ne compte que les catégories ordinaires. Utilisez la propriété `SpecialCategoryCount` pour comptabiliser les catégories racine, les catégories de branche et les catégories spéciales.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

### Accès

Lecture

### Exemples

```
objDateDrillDown = objModel.Dimensions("Years").DrillDowns(1)objLevel  
= objDateDrillDown.Levels("Year") For intX = 1 To objLevel.CategoryCount
```

---

## Propriété `CategoryCountLevel`

La propriété `CategoryCountLevel` définit ou retourne l'objet de niveau auquel s'applique une mesure de nombre de catégories.

### Syntaxe

*Measure*.`CategoryCountLevel`

### Application

Objet Measure

### Explication

Il est possible de définir une mesure dénombrant les catégories. Par exemple, vous pouvez créer une mesure qui indique le nombre de clients de chaque type ayant acheté un produit donné chaque mois, chaque trimestre ou chaque année.

Cette propriété permet de spécifier le niveau auquel s'applique une mesure de nombre de catégories. Transformer exclut du nombre les catégories non manquantes ou non nulles. Le nombre de catégories devient la valeur numérique de la mesure.

Une fois que cette propriété est définie, la propriété `MeasureType` retourne une valeur de la constante `trCountMeasure`. Une mesure de nombre de catégories dérive des valeurs de la comptabilisation des catégories et ne peut pas utiliser une association ou une expression pour la dérivation des valeurs.

Assurez-vous que la propriété `Unique` du niveau ou du niveau de date a pour valeur `True` avant de définir cette propriété.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet

### Accès

Lecture/écriture

### Exemples

```
objMeasure.CategoryCountLevel = objLevelobjMeasure.CategoryCountLevel  
= objLevel
```

---

## Propriété CategorySets

La catégorie CategorySets retourne une collection des objets CategorySet associés à une définition de calcul.

### Syntaxe

*CalculationDefinition* .CategorySets

### Application

Objet CalculationDefinition

### Explication

Un objet CategorySet contient une ou plusieurs catégories utilisées dans un calcul.

Quand vous créez une définition de calcul, certaines fonctions de calcul permettent de spécifier un ou plusieurs ensembles de catégories comme paramètres. La fonction de pourcentage de croissance, par exemple, calcule la différence en pourcentage entre le deuxième et le premier paramètre.

### Type

Objet - Category Sets

### Accès

Lecture

### Exemples

```
catSet = calcDef.CategorySets.Add()
```

---

## Propriété CharacterType

### Description

La propriété CharacterType définit ou retourne le type du jeu de caractères utilisé par la source de données.

## Syntaxe

*FlatFileDataSource* .CharacterType

## Application

Objet FlatFileDataSource

## Explication

Cette propriété permet de définir le jeu de caractères Windows ANSI (ISO 8859-1) ou DOS (OEM).

Nous vous recommandons d'utiliser le même jeu de caractères sur le client et le serveur pendant que vous développez le modèle client-serveur afin de garantir des résultats corrects avec les caractères étendus (supérieurs à 128).

La propriété CharacterType utilise les valeurs de la constante xtrCharacterType.

## Type

Constante - xtrCharacterType

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources.
  Add(xtrObjectType.trFlatFileDataSource)objDataSource.CharacterType
= xtrCharacterType.trCharAnsiSingleByte
```

---

## Propriété ChildCategories

La propriété ChildCategories renvoie une collection Categories.

## Syntaxe

*object* .ChildCategories

## Application

Objet Category

Objet SpecialCategory

## Explication

Chaque objet Category peut retourner une collection Categories à partir d'un niveau inférieur au moyen de la propriété ChildCategories.

## Type

Objet - Categories



## Accès

Lecture

## Exemples

```
objCategory  
= objModel.Dimensions("Retailers").DrillDowns(1).Categories(2)objCatToMove  
= objCategory.ChildCategories(5)
```

---

## Propriété ChildCubes

La propriété ChildCubes retourne une collection d'objets ChildCube à partir d'un groupe de cubes.

## Syntaxe

*CubeGroup* .ChildCubes

## Application

Objet CubeGroup

## Explication

Un groupe de cubes représente un niveau dans une dimension. Chaque cube enfant de la collection reflète une catégorie unique dans ce niveau et inclut toutes les catégories descendantes de cette catégorie.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - ChildCubes

## Accès

Lecture

## Exemples

```
objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)objChildCube =  
objCubesByRegion.ChildCubes("Central Europe")
```

---

## Propriété ChildCustomViews

La propriété ChildCustomViews renvoie une collection d'objets CustomView.

## Syntaxe

*CustomView* .ChildCustomViews

## Application

Objet CustomView

## Explication

Utilisez cette propriété pour créer une vue personnalisée imbriquée dans une autre vue personnalisée. La méthode Add de la collection CustomViews permet d'ajouter une nouvelle vue personnalisée. Après avoir ajouté la vue personnalisée à la collection, vous pouvez mettre à jour les propriétés du nouvel objet CustomView.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

CustomViews

## Accès

Lecture

## Exemples

```
currentCustomView = model.CustomViews.Add()childCustomView  
= currentCustomView.ChildCustomViews.Add()
```

---

## Propriété ChildMeasures

La propriété ChildMeasures renvoie une collection de mesures enfant qui proviennent du dossier de mesures.

## Syntaxe

*Measure* .ChildMeasures

## Application

Objet Measure

## Explication

Utilisez cette propriété pour extraire toutes les mesures enfants. La collection retournée est un objet MeasureCollection avec les mêmes propriétés et méthodes. Si vous utilisez cette propriété en conjonction avec une mesure ordinaire, une exception se produira.

Renvoie une collection vide si un dossier de mesures ne contient aucune mesure enfant. Une exception COM est émise avec le message "trMeasure (ChildMeasures) : Cette opération n'est pas autorisée pour cet objet".

L'objet MeasureCollection obtenu de l'objet Model contient aussi le dossier de mesures et les mesures enfants. Il est possible d'accéder ces dossiers et à leurs enfants de la même manière que tout autre objet Measurecollection; cependant, un objet MeasureCollection transmis par un objet Model génère une erreur si la méthode Move a été appliquée à un dossier de mesures ou à ses mesures enfants.

## Type

Objet - Categories

## Accès

Lecture

---

## Propriété Code

La propriété Code définit ou retourne un code qui identifie de manière unique la catégorie dans toute la dimension.

### Syntaxe

*object* .Code

### Application

Objet Category

Objet SpecialCategory

### Explication

Le code est un identificateur alphabétique ou alphanumérique interne qui est défini automatiquement quand Transformer génère des catégories. Vous pouvez le remplacer par un code plus significatif. Si vous spécifiez un code qui existe déjà, Transformer lui ajoute un suffixe numérique pour le rendre unique.

### Type

Chaîne

## Accès

Lecture/écriture

### Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories
  .Add(xtrObjectType.trSpecialCategory)objSpecCategory.Code
= "Five Month Period"
```

---

## Propriété Columns

La propriété Columns retourne une collection Columns qui contient tous les objets Column d'une source de données.

### Syntaxe

*object* .Columns

### Application

Objet CrossTabDataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

Objet Query

## Explication

Les colonnes constituent la source de toutes les données utilisées par Transformer quand vous créez un cube.

Utilisez la propriété ColumnsLoaded pour déterminer si les données de colonne ont servi à créer les dimensions, les niveaux et les mesures d'un modèle.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture

## Exemples

```
objDataSource  
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objColumn  
= objDataSource.Columns(1)
```

---

## Propriété ColumnsLoaded

La propriété ColumnsLoaded retourne une valeur indiquant si les colonnes de la source de données ont été utilisées pour créer le modèle.

## Syntaxe

*object*.ColumnsLoaded

## Application

Objet CrossTabDataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

La valeur de retour True indique que des objets Column du modèle sont liés à des colonnes de la source de données sous-jacente.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Booléen

### **Accès**

Lecture

---

## **Propriété CompressMDC**

La propriété CompressMDC définit ou renvoie une valeur indiquant si un cube est comprimé.

### **Syntaxe**

*object* .CompressMDC

### **Application**

Objet Cube

Objet CubeGroup

### **Explication**

Cette propriété permet d'optimiser la distribution d'un cube.

Transformer compresse le cube chaque fois qu'il est créé. Quand vous effectuez une opération de Transformer sur un cube compressé, notamment la mise à jour incrémentielle ou la vérification du statut du cube, cette application décompresse le cube, termine l'action, puis compresse à nouveau le cube.

La première ouverture d'un cube compressé dans PowerPlay peut durer un certain temps. Ensuite, le cube retrouve sa taille d'origine et s'ouvre normalement.

**Paramètre par défaut** : False

### **Type**

Booléen

### **Accès**

Lecture/écriture

### **Exemples**

```
objCube = objModel.Cubes.Item(1)objCube.CompressMDC =  
False
```

---

## **Propriété Connection**

La propriété Connection définit ou retourne la connexion à Content Manager.

## Syntaxe

*PackageDatasourceConnection* .**Connection**

## Application

Objet PackageDatasourceConnection

## Explication

Utilisez cette propriété pour extraire la connexion à Content Manager.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
package = model.Packages.Add()connection  
= package.PackageDatasourceConnections.Add()connection.Connection  
= "great_outdoors_warehouse"
```

---

## Propriété Consolidate

La propriété Consolidate définit ou renvoie une valeur indiquant comment un cube est consolidé.

## Syntaxe

*object* .**Consolidate**

## Application

Objet ChildCube

Objet Cube

Objet CubeGroup

## Explication

Cette propriété permet de réduire la taille du cube et le temps d'accès dans PowerPlay. Un cube consolidé cumule en un seul enregistrement les valeurs identiques qui ne sont pas des mesures, et récapitule les valeurs de mesure.

Cette propriété utilise les valeurs de la constante xtrCubeConsolidate.

Les enregistrements de données contiennent des valeurs de non-mesures identiques lorsque :

- la source contient des transactions dont les valeurs de non-mesures sont identiques, par exemple lorsque le même client achète deux fois le même produit le même jour ;

- le degré de détail le permet, notamment quand la propriété `DateDegreeofDetail` associée a pour valeur `Month`, les valeurs du jour des transactions sources sont ignorées pendant la consolidation des cubes ;
- une dimension est omise à partir du cube, notamment quand deux ventes du même produit sont effectuées dans des magasins différents le même jour. Si les magasins sont omis à partir du cube, les enregistrements des ventes possèdent des valeurs de non-mesures identiques ;
- les catégories du cube sont récapitulées ou supprimées, par exemple lorsque deux ventes d'un même produit sont effectuées auprès du même client, le même jour, mais que les couleurs diffèrent. Si les couleurs sont omises du cube par l'emploi de l'une ou l'autre option de la vue de dimension, les enregistrements des ventes possèdent des valeurs de non-mesures identiques.

Dans les trois premiers cas décrits ci-dessus, la consolidation utilise la valeur de la propriété `DuplicateRollup` (si elle diffère de la valeur `None`) pour combiner des enregistrements possédant des valeurs identiques dans leurs colonnes de non-mesures. Dans le dernier exemple, à moins que la valeur de la propriété `TimeStateRollup` soit différente de `None` pour la mesure, la consolidation utilise la valeur de la propriété `RegularRollup` pour combiner des enregistrements pour lesquels les vues de la dimension donnent des valeurs identiques.

## Type

Constante - `xtrCubeConsolidate`

## Accès

Lecture/écriture

## Exemples

```
objCube.Consolidate = xtrCubeConsolidate.trConsolidateDefault
```

---

## Propriété Context

La propriété `Context` définit ou retourne la branche de niveaux utilisée pour classer des catégories au sein d'un niveau.

### Syntaxe

*Association* .`Context`

### Application

Objet `Association`

### Explication

Par défaut, des catégories apparaissent dans un niveau, puis dans un cube selon leur ordre d'apparition dans la source de données. Pour trier ces objets, utilisez un objet `Association` et affectez à la propriété `AssociationRole` la valeur `trAssociationOrderBy`.

Utilisez la propriété Context pour nommer la branche de niveaux à laquelle s'applique le tri. Quand un niveau représente la convergence de deux branches de niveaux ou davantage, vous pouvez appliquer un ordre de tri différent à chacune d'elles.

Utilisez la propriété OrderByDescending pour spécifier le classement du tri. Si vous définissez le tri à l'aide d'une colonne de données numériques, spécifiez la taille du type de données à l'aide de OrderByStorageType.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()objAssociation.Context  
= objDrill
```

---

## Propriété ContextLevel

La propriété ContextLevel définit ou renvoie une période.

## Syntaxe

*SpecialCategory* .ContextLevel

## Application

Objet SpecialCategory

## Explication

Créez une catégorie spéciale pour effectuer le suivi des mesures d'une période de date spécifique par rapport à la date en cours. Utilisez les propriétés ContextLevel et ContextOffset pour définir des catégories de date relative telles que l'année précédente ou le trimestre précédent.

Cette propriété permet de spécifier le contexte de la propriété TargetLevel ou ToDateLevel. Si la période cible est le mois, par exemple, la période de contexte est l'année ou le trimestre.

## Type

Chaîne

## Accès

Lecture/écriture



## Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories
  .Add(xtrObjectType.trSpecialCategory)objSpecCategory.ContextLevel
= "Quarter"
```

---

## Propriété ContextOffset

La propriété ContextOffset définit ou retourne la position de la période de référence par rapport à la période en cours.

### Syntaxe

*SpecialCategory* .ContextOffset

### Application

Objet SpecialCategory

### Explication

Cette propriété permet de spécifier une valeur de décalage, par rapport à la période de date en cours, pour la propriété ContextLevel. Si la période de contexte est l'année, par exemple, la valeur de décalage du contexte égale à -1 correspond à la première année antérieure.

Dans le cas d'une période de date groupée ou totalisée, la propriété ContextOffset spécifie le décalage du groupe des catégories spéciales.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Court

### Accès

Lecture/écriture

### Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories
  .Add(xtrObjectType.trSpecialCategory)objSpecCategory.ContextOffset
= -3
```

---

## Propriété ConvergenceLevel

La propriété ConvergenceLevel retourne le niveau auquel se joignent ou convergent des branches de niveaux principale et secondaire.

### Syntaxe

*object* .ConvergenceLevel

## Application

Objet DateDrillDown

Objet DrillDown

## Explication

Vous pouvez configurer des structures de branches de niveaux secondaires au sein d'une dimension pour fournir différentes perspectives sur les données dans PowerPlay. Chaque branche de ce type se connecte à la branche principale au niveau de convergence. Chaque catégorie d'un niveau de convergence doit être unique et non ambiguë.

Quand un niveau représente la convergence de deux branches de niveaux ou davantage, vous pouvez appliquer un ordre de tri différent à chacune d'elles. Chaque tri requiert un objet Association dont la propriété Context a pour valeur la branche de niveaux de votre choix.

Pour créer une branche de niveaux secondaire, utilisez la méthode CreateAlternateDrillDown.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()
```

---

## Propriété Count

La propriété Count renvoie le nombre d'objet dans la collection.

## Syntaxe

*collection* .Count

## Application

Collection Associations

Collection CalculationDefinitions

Collection Categories

Collection CategorySets

Collection ChildCubes  
Collection Columns  
Collection CubeCustomViews  
Collection Cubes  
Collection CurrencyRates  
Collection CurrencyRecords  
Collection CurrencyTables  
collection CustomViews  
Collection DataSources  
Collection DimensionLevels  
Collection Dimensions  
Collection DrillDowns  
Collection DrillThroughTargets  
Collection Filters  
collection LevelCategories  
Collection LevelDrillDowns  
Collection Levels  
Collection Measures  
Collection Names  
Collection Namespaces  
Collection PackageDatasourceConnections  
Collection Packages  
Collection Prompts  
Collection Queries  
Collection Reports  
Collection SecurityObjects  
Collection Signons  
Collection SuspendedModels

Collection Views

### **Explication**

La valeur zéro indique que la collection ne contient pas d'objet.

### **Type**

Long

### **Accès**

Lecture

### **Exemples**

```
For intX = 1 To objModel.CheckModel.Count
```

---

## **Propriété CountryCode**

La propriété CountryCode définit ou renvoie un code de pays ou de région.

### **Syntaxe**

*CurrencyRecord* .CountryCode

### **Application**

Objet CurrencyRecord

### **Explication**

Transformer utilise le pays spécifié dans les paramètres régionaux du Panneau de configuration de Windows pour déterminer la valeur par défaut des attributs de devise tels que le symbole monétaire et le nombre de positions décimales. Utilisez la propriété CountryCode pour remplacer ces valeurs par défaut.

Par défaut, le code de pays ou de région est un code ISO-3166 à trois caractères qui spécifie le pays dans lequel est utilisée la devise, notamment ITA pour Italie. Si vous le souhaitez, vous pouvez remplacer ces codes.

### **Type**

Chaîne

### **Accès**

Lecture/écriture

### **Exemples**

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRecord.CountryCode  
= "AUS"
```

---

## Propriété CubeCodePage

La propriété CubeCodePage définit ou retourne le paramètre de page de codes du cube du modèle utilisé pour créer le cube. Une chaîne vide est renvoyée si le modèle est associé au paramètre par défaut qui correspond à la page de codes du système d'exploitation.

### Syntaxe

*Model* .CubeCodePage

### Application

Objet Model

### Explication

Utilisez cette propriété afin d'extraire ou de modifier le paramètre de page de codes du cube de traitement pour un modèle. Si vous modifiez le paramètre de page de codes de traitement du modèle après avoir créé les cubes, vous devrez supprimer les cubes créés avec la page de codes initiale et les recréer avec le nouveau paramètre de page de codes. Seules les chaînes valides sont acceptées. Si une valeur incorrecte est affectée, elle est redéfinie sur le paramètre par défaut. Les valeurs actuellement prises en charge sont les suivantes :

- ANSI\_X3.4-1968
- Big5
- EUC-CN
- EUC-KR
- Extended\_UNIX\_Code\_Packed\_Format\_for\_Japanese
- GB\_2312
- GB\_2312-80
- GB18030
- GBK
- hp-roman8
- IBM850
- ISO\_8859-1
- ISO\_8859-1:1987
- ISO\_8859-2:1987
- ISO\_8859-3:1988
- ISO\_8859-4:1988
- ISO\_8859-5:1988
- ISO\_8859-6:1987
- ISO\_8859-7:1987
- ISO\_8859-8:1988
- ISO\_8859-9:1989
- ISO\_8859-10
- ISO\_8859-11
- ISO\_8859-13
- ISO\_8859-14

- ISO\_8859-15
- KS\_C\_5601-1987
- Shift\_JIS
- TIS-620
- UTF-8
- windows-1250
- windows-1251
- windows-1252
- windows-1253
- windows-1254
- windows-1255
- windows-1256
- windows-1257
- windows-874
- windows-936

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété CubeCreation

La propriété CubeCreation définit ou retourne une valeur indiquant si le cube est créé.

### Syntaxe

*object* .CubeCreation

### Application

Objet ChildCube

Objet Cube

Objet CubeGroup

### Explication

Utilisez cette propriété avec la propriété Status en tant que filtre pour sélectionner un sous-ensemble des cubes disponibles pour la création. Vous pouvez, par exemple, sélectionner tous les cubes pour lesquels des avertissements sont émis.

En règle générale, ces deux propriétés sont utilisées pour sélectionner tous les cubes, des cubes qui ont été créés précédemment sans erreur ou des cubes dont la création a posé problème auparavant.

La propriété CubeCreation utilise les valeurs de la constante xtrCubeCreation.

## Type

Constante - xtrCubeCreation

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion  
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objChildCube.CubeCreation  
= xtrCubeCreation.trCubeCreationON
```

---

## Propriété CubeCustomViews

La propriété CubeCustomViews renvoie une collection d'objets CustomView.

## Syntaxe

*object*.CubeCustomViews

## Application

Objet ChildCube

Objet Cube

Objet CubeGroup

## Explication

Cette propriété permet d'associer une vue personnalisée à un cube. Vous devez d'abord employer les propriétés de l'objet CustomView pour définir une vue personnalisée. Utilisez ensuite la méthode Add pour ajouter un objet CustomView à la collection CubeCustomViews du cube.

Chaque objet Cube, CubeGroup ou ChildCube définit une collection CubeCustomViews.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CubeCustomViews

## Accès

Lecture

## Exemples

```
cube = model.Cubes.Item("Sales and Marketing")cube.CubeCustomViews.Add(custom_view)
```

---

## Propriété Cubes

La propriété Cubes renvoie une collection d'objets Cube et CubeGroup.

### Syntaxe

*Model* .Cubes

### Application

Objet Model

### Discussion

La collection n'inclut pas les cubes enfants. Pour retourner les cubes enfants à partir d'un groupe de cubes, utilisez la propriété ChildCubes.

### Type

Objet - Cubes

### Accès

Lecture

### Exemples

```
objCube = objModel.Cubes.Item(1)
```

---

## Propriété CubeStamp

La propriété CubeStamp renvoie un horodatage de création de cube.

### Syntaxe

*object* .CubeStamp

### Application

Objet Cube

Objet CubeGroup

### Explication

La valeur de la propriété CubeStamp est définie de manière interne. Si cette valeur ne concorde pas avec celle du modèle, Transformer définit le statut du PowerCube comme non valide.

Un résultat non valide peut indiquer que le cube a été créé à partir d'un autre modèle ou d'une version obsolète du modèle en cours.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



## Type

Long

## Accès

Lecture

---

## Propriété CurrencyCountryLabel

La propriété CurrencyCountryLabel définit ou retourne le libellé du pays ou de la région de la devise.

## Syntaxe

*CurrencyRecord* .CurrencyCountryLabel

## Application

Objet CurrencyRecord

## Explication

Il s'agit d'un nom descriptif pour indiquer le nom de la devise.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété CurrencyDecimals

La propriété CurrencyDecimals définit ou retourne le nombre de décimales utilisées dans une devise.

## Syntaxe

*CurrencyRecord* .CurrencyDecimals

## Application

Objet CurrencyRecord

## Explication

Transformer utilise le pays spécifié dans les paramètres régionaux du Panneau de configuration de Windows pour déterminer la valeur par défaut des décimales de la devise. La valeur des décimales peut changer quand vous définissez la propriété CountryCode.

Utilisez la propriété `CurrencyDecimals` pour définir le nombre de décimales. La valeur de cette propriété a la priorité sur les paramètres régionaux et la propriété `CountryCode`.

Vous devez affecter la valeur `True` à la propriété `CurrencyFormatOverride` avant de pouvoir modifier la propriété `CurrencyDecimals`.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRecord.CurrencyDecimals  
= 2
```

---

## Propriété `CurrencyFormatOverride`

La propriété `CurrencyFormatOverride` définit ou retourne une valeur indiquant si vous pouvez écraser les attributs de devise.

## Syntaxe

*CurrencyRecord* .**CurrencyFormatOverride**

## Application

Objet `CurrencyRecord`

## Explication

Transformer utilise le pays spécifié dans les paramètres régionaux du Panneau de configuration de Windows pour déterminer la valeur par défaut des attributs de devise tels que le symbole monétaire et le nombre de décimales. Vous pouvez définir ces attributs à l'aide de la propriété `CountryCode`. Pour écraser un attribut de devise de manière spécifique avec les propriétés `CurrencyDecimals` ou `CurrencySymbol`, vous devez d'abord affecter la valeur `True` à la propriété `CurrencyFormatOverride`.

Vous pouvez écraser des attributs de devise :

- lorsque sur l'ordinateur de l'utilisateur de PowerPlay, aucun pays n'est défini dans les paramètres régionaux du Panneau de configuration de Windows ou lorsqu'un pays utilise plusieurs devises.
- pour ajouter un symbole monétaire descriptif. Vous pouvez, par exemple, ajouter au symbole du dollar le préfixe `AUS`, `CDN` ou `US` pour différencier chaque devise qui est affichée dans PowerPlay.

**Valeur par défaut :** `False`

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord  
= objModel.CurrencyRecords.Add()objCurrencyRecord.CurrencyFormatOverride  
= True
```

---

## Propriété CurrencyIsEMU

La propriété CurrencyIsEMU définit ou retourne une valeur indiquant si l'enregistrement est un enregistrement de devise de l'UEM (Union économique et monétaire).

## Syntaxe

*CurrencyRecord*.CurrencyIsEMU

## Application

Objet CurrencyRecord

## Explication

Les devises de l'UEM nécessitent une table de l'euro distincte de la table de base. Utilisez cette propriété pour ajouter un enregistrement de devise à la table de l'euro. Quand vous définissez cette propriété, vous devez aussi définir la propriété EMUEntryDate.

Si la conversion s'effectue entre une devise de l'UEM et une devise non UEM, Transformer utilise un processus de triangulation. Transformer effectue la conversion entre :

- la devise source et la devise euro de base,
- la devise euro de base et la devise source,
- la devise de base et la devise cible.

Le processus inverse est appliqué quand Transformer convertit une devise non UEM en une devise UEM.

Quand vous créez une table de l'euro, vous devez utiliser la méthode LoadCurrencyTable à différentes étapes du processus. Utilisez cette méthode après avoir :

- associé chaque colonne de la source de données à un rôle d'association dans la table de l'euro,
- ajouté des enregistrements de devise à la table de l'euro.

**Paramètre par défaut** : False

## Type

Valeur booléenne

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRec = objModel.CurrencyRecords(1)objCurrencyRec.CurrencyIsEMU  
= True
```

---

## Propriété CurrencyIsEuro

La propriété CurrencyIsEuro définit ou retourne une valeur indiquant si l'enregistrement est la devise euro de base.

## Syntaxe

*CurrencyRecord* .**CurrencyIsEuro**

## Application

Objet CurrencyRecord

## Explication

Cette propriété permet d'identifier un enregistrement de devise comme devise euro de base dans la collection CurrencyRecords. La conversion vers ou à partir de l'euro requiert une valeur de base par rapport à laquelle sont fixés tous les autres taux de conversion dans la table de l'euro.

Quand vous ajoutez une table de l'euro à la collection CurrencyTables, Transformer affecte automatiquement la valeur True à la propriété CurrencyIsEuro de l'enregistrement <base Euro>. Tous les autres enregistrements retournent la valeur False.

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRecord.CurrencyIsEuro  
= False
```

---

## Propriété CurrencyRates

La propriété CurrencyRates renvoie une collection d'objets CurrencyRates.

## Syntaxe

*CurrencyRecord* .CurrencyRates

## Application

Objet CurrencyRecord

## Explication

Les objets CurrencyRate représentent des taux de conversion monétaire au sein d'un objet CurrencyRecord.

Chaque objet stocke un taux de conversion défini au niveau de détail d'une date telle que l'année, le trimestre, le mois, la semaine ou le jour. Utilisez la propriété Category de l'objet CurrencyRate pour retourner la catégorie de date associée à un taux de conversion spécifique.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CurrencyRates

## Accès

Lecture

## Exemples

```
objCurrencyRate = objCurrencyRecord.CurrencyRates(intX)
```

---

## Propriété CurrencyRecord

La propriété CurrencyRecord renvoie l'objet CurrencyRecord auquel s'applique l'objet CurrencyRate.

## Syntaxe

*CurrencyRate* .CurrencyRecord

## Application

Objet CurrencyRate

## Explication

Chaque objet CurrencyRecord représente une ligne d'informations relatives à la conversion monétaire dans une table des devises. Utilisez la propriété CountryCode de l'objet CurrencyRecord, par exemple, pour retourner le code de pays ou de région auquel sont associés l'enregistrement et le taux de conversion.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CurrencyRecord

## Accès

Lecture

---

## Propriété CurrencyRecords

La propriété CurrencyRecords retourne une collection de tous les objets CurrencyRecord dans un modèle.

## Syntaxe

*Model* .CurrencyRecords

## Application

Objet Model

## Discussion

Chaque objet CurrencyRecord représente une ligne d'informations relatives à la conversion monétaire dans un objet CurrencyTable.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CurrencyRecords

## Accès

Lecture

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()
```

---

## Propriété CurrencySymbol

La propriété CurrencySymbol définit ou retourne le symbole monétaire utilisé par une devise.

## Syntaxe

*CurrencyRecord* .CurrencySymbol

## Application

Objet CurrencyRecord

## Explication

Transformer utilise le pays ou la région spécifiés dans les paramètres régionaux du Panneau de configuration de Windows pour déterminer la valeur par défaut du symbole monétaire. En règle générale, le symbole change quand vous définissez la propriété `CountryCode`.

La propriété `CurrencySymbol` permet de définir le symbole monétaire de manière spécifique et d'écraser les paramètres régionaux ainsi que la propriété `CountryCode`. Vous pouvez, par exemple, ajouter au symbole du dollar le préfixe `AUS`, `CDN` ou `US` pour différencier chaque devise qui est affichée dans `PowerPlay`.

Vous devez affecter la valeur `True` à la propriété `CurrencyFormatOverride` avant de pouvoir modifier la propriété `CurrencySymbol`.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord.CurrencySymbol = "$"
```

---

## Propriété `CurrencyTable`

La propriété `CurrencyTable` renvoie un objet `CurrencyTable` relatif à un taux de conversion.

## Syntaxe

```
CurrencyRate.CurrencyTable
```

## Application

Objet `CurrencyRate`

## Explication

L'objet `CurrencyTable` contient les enregistrements de conversion monétaire utilisés pour la conversion monétaire. Dans `PowerPlay`, les valeurs de mesure spécifiques à la devise sont affichées dans les devises que vous définissez, au niveau de détail de la date pour lequel les taux de conversion sont disponibles.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - `CurrencyTable`

## Accès

Lecture

---

## Propriété CurrencyTables

La propriété CurrencyTables renvoie une collection d'objets CurrencyTable.

### Syntaxe

*Model* .CurrencyTables

### Application

Objet Model

### Explication

Cette collection permet d'ajouter, de sélectionner ou de supprimer un objet CurrencyTable. Transformer utilise les informations de la table des devises d'un objet CurrencyTable pour effectuer les conversions appropriées quand les utilisateurs affichent des cubes dans PowerPlay.

Les conversions de devises de l'UEM exigent deux tables. Dans une table, la propriété CurrencyTableType a pour valeur trCurrencyTableBase et dans l'autre, trCurrencyTableEuro. Une collection CurrencyTables ne peut contenir qu'un seul type de table.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet - CurrencyTables

## Accès

Lecture

### Exemples

```
objBaseTable = objModel.CurrencyTables.Add()
```

---

## Propriété CurrencyTableType

La propriété CurrencyTableType définit ou retourne le type de table des devises.

### Syntaxe

*CurrencyTable* .CurrencyTableType

### Application

Objet CurrencyTable



## Explication

Utilisez cette propriété pour spécifier si le type de la table des devises est :

- base, une table des devises liée à votre devise locale telle qu'elle est déterminée par les paramètres régionaux du Panneau de configuration de Windows
- euro, une table de devises de l'UEM (Union économique et monétaire) définies avec l'euro au 1er janvier 1999,
- autre, une table des devises personnalisées.

Les conversions de devises de l'UEM exigent deux tables. Dans une table, la propriété `CurrencyTableType` a pour valeur `trCurrencyTableBase` et dans l'autre, `trCurrencyTableEuro`. Une collection `CurrencyTables` ne peut contenir qu'un seul type de table.

La propriété `CurrencyTableType` utilise les valeurs de la constante `xtrCurrencyTableType`.

## Type

Constante - `xtrCurrencyTableType`

## Accès

Lecture/écriture

## Exemples

```
objBaseTable = objModel.CurrencyTables.Add()objBaseTable.CurrencyTableType  
= xtrCurrencyTableType.trCurrencyTableBase
```

---

## Propriété CurrentModel

La propriété `CurrentModel` retourne l'objet `Model` qui est actuellement ouvert dans `Transformer`.

## Syntaxe

*Application* .`CurrentModel`

## Application

Objet `Application`

## Explication

Utilisez cette propriété si vous devez référencer le modèle actif dans votre script.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - `Model`

## Accès

Lecture

---

## Property CurrentValueIndex

La propriété CurrentValueIndex définit ou retourne l'index de valeur d'invite en cours.

### Syntaxe

*Prompt* .CurrentValueIndex

### Application

Objet Prompt

### Explication

Cette propriété permet d'effectuer une itération dans les valeurs d'invite pour les invites à plusieurs valeurs. L'index commence à 1.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Entier

### Accès

Lecture/écriture

### Exemples

```
new_report = model.Reports.Add()new_query = new_report.Queries.Add()new_prompt  
= new_query.Prompts.Add()new_prompt.CurrentValueIndex = index
```

---

## Propriété CustomView

La propriété CustomView définit ou retourne une valeur indiquant si une vue contient toutes, quelques-unes ou aucune des catégories d'une dimension.

### Syntaxe

*View* .CustomView

### Application

Objet View

### Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CustomView

## Accès

Lecture/écriture

---

## Propriété CustomViews

La propriété CustomViews renvoie une collection d'objets CustomView.

## Syntaxe

*model* .CustomViews

## Application

Objet View

Objet Model

## Explication

Cette propriété permet de créer une vue personnalisée dans un modèle. La méthode Add de la collection CustomViews permet d'ajouter une nouvelle vue personnalisée. Après avoir ajouté la vue personnalisée à la collection, vous pouvez mettre à jour les propriétés du nouvel objet CustomView.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - CustomViews

## Accès

Lecture

## Exemples

```
custom_view = model.CustomViews.Add()
```

---

## Propriété DataCharacterSet

La propriété DataCharacterSet définit ou retourne le jeu de caractères par défaut utilisé par l'application.

## Syntaxe

*Application* .DataCharacterSet

## Application

Objet Application

## Explication

Cette propriété permet de modifier le jeu de caractères. Transformer reconnaît les jeux de caractères Windows ANSI (ISO 8859-1) et DOS (OEM).

Nous vous recommandons d'utiliser le même jeu de caractères sur le client et le serveur pendant que vous développez le modèle client-serveur afin de garantir des résultats corrects avec les caractères étendus (supérieurs à 128).

La propriété DataCharacterSet utilise les valeurs de la constante xtrCharacterType.

## Type

Constante - xtrCharacterType

## Accès

Lecture/écriture

## Exemples

```
objTransApp.DataCharacterSet = xtrCharacterType.trCharDefault
```

---

## Propriété DataClass

La propriété DataClass définit ou retourne le type de données de la colonne source.

## Syntaxe

*Column* .DataClass

## Application

Objet Column

## Explication

La définition de la valeur de cette propriété permet d'accélérer les actions de la méthode DoAutoDesign. En règle générale, les types de classe de données numérique, de date et de texte sont utilisés pour décrire respectivement les données de performance, de date et de structure.

Quand vous référencez une colonne dans la propriété ExpressionText, sa propriété DataClass doit avoir une valeur d'entrée valide sous peine de générer une erreur d'exécution.

La propriété DataClass utilise les valeurs de la constante xtrDataClass.

## Type

Constante - xtrDataClass

## Accès

Lecture/écriture

## Exemples

```
objDataSource  
= objModel.DataSources.  
  Add(xtrObjectType.trFlatFileDataSource)objDataSource.DataClass  
= xtrDataClass.trDataClassDate
```

---

## Propriété DataRange

La propriété DataRange définit ou retourne le nom d'une plage de base de données.

## Syntaxe

*DbDataSource* .DataRange

## Application

Objet DbDataSource

## Explication

Utilisez cette propriété pour spécifier le nom d'une table ou d'un bloc de données comme défini dans le fichier source de la base de données.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété DataSource

La propriété DataSource définit ou retourne la source de données dans Content Manager.

## Syntaxe

*PackageDatasourceConnection* .DataSource

## Application

Objet PackageDatasourceConnection

## Explication

Utilisez cette propriété pour obtenir ou définir la source de données figurant dans Content Manager.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété DataSourcePath

La propriété DataSourcePath définit ou retourne l'emplacement dans lequel Transformer recherche les fichiers de la source de données.

## Syntaxe

*Application* .DataSourcePath

## Application

Objet Application

## Explication

Lorsque aucun emplacement n'est spécifié, Transformer recherche le dossier d'installation de PowerPlay.

Utilisez la propriété LocalPath pour spécifier le chemin d'accès et le nom de fichier de chaque source de données. Si la propriété LocalPath spécifie un chemin d'accès relatif, il est ajouté à la valeur définie dans la propriété DataSourcePath. Dans ce cas, les valeurs concaténées doivent donner comme résultat un emplacement correctement défini.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objTransApp.DataSourcePath = strStartLocation
```

---

## Propriété DataSources

La propriété DataSources retourne une collection d'objets DataSource dans un modèle.

## Syntaxe

*Model* .DataSources

## Application

Objet Model

## Explication

Utilisez cette collection pour regrouper toutes les sources de données dans un modèle. Ce groupe inclut des objets génériques tels que DataSource ainsi que des objets sources spécifiques, comme IqdDataSource.

Les sources de données, à l'exception de l'objet DataSource, contiennent des collections Columns. Les colonnes correspondent à la source des niveaux dans une dimension.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture

## Exemples

```
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
```

---

## Propriété DataSourceWindowsLocation

La propriété DataSourceWindowsLocation définit ou renvoie l'emplacement du cube, ce qui comprend le chemin d'accès complet et le nom du cube.

## Syntaxe

*Cube*.DataSourceWindowsLocation

## Application

Objet Cube

## Explication

Cette propriété permet de définir l'emplacement du cube depuis le cadre de référence du serveur IBM Cognos BI.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
cube = model.Cubes.Item("National")cube.DataSourceWindowsLocation  
= "c:\NATIONAL\Deployment1\National.mdc"
```

---

## Propriété DataTemporaryFilePath

La propriété DataTemporaryFilePath définit ou retourne l'emplacement dans lequel Transformer crée des fichiers de travail temporaires lorsqu'il génère des cubes.

### Syntaxe

*Application* .DataTemporaryFilePath

### Application

Objet Application

### Explication

Utilisez cette propriété pour spécifier un ou plusieurs emplacements pour le traitement des cubes extrêmement volumineux lorsque l'espace disque d'un seul emplacement est insuffisant pour tous les fichiers. Délimitez les répertoires par un point virgule (;).

Lorsqu'aucun emplacement n'est spécifié, Transformer recherche l'emplacement (dans l'ordre suivant) :

- spécifié par la propriété ModelTemporaryFilePath,
- spécifié par la propriété ModelsPath,
- spécifié par la propriété PowerCubesPath,
- du répertoire d'installation de PowerPlay
- défini par la variable Temporary = dans la section [Services] du fichier cs7g.ini
- défini par la variable d'environnement TEMP,
- défini par la variable d'environnement TMP.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objTransApp.DataTemporaryFilePath = strStartLocation
```

---

## Propriété DateDegreeofDetail

La propriété DateDegreeofDetail définit ou retourne le niveau de date auquel des rapports peuvent être générés.



## Syntaxe

*Column* .DateDegreeofDetail

## Application

Objet Column

## Explication

Utilisez cette propriété pour définir le niveau de date auquel des rapports peuvent être générés dans Transformer. Le niveau de date peut être le jour, la semaine, le mois, le trimestre ou l'année.

Quand un niveau est défini, le rapport des mesures dans la dimension est limité à ce niveau de détail. Le niveau ne peut pas être inférieur à l'entrée de date définie par la propriété DateInputFormat. Si l'entrée de date est le trimestre, par exemple, vous ne pouvez pas définir le mois comme degré de détail.

La propriété DateDegreeofDetail utilise les valeurs de la constante xtrDateLevel. Si vous affectez la constante trDateLevelUnspecified, Transformer définit le degré de détail sur base d'autres attributs de la colonne source.

## Type

Constante - xtrDateLevel

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objDataSource
.Columns("EuroDate").DateDegreeofDetail
= xtrDateLevel.trDateLevelMonth
```

---

## Propriété DateDegreeofDetailLevelName

La propriété DateDegreeofDetailLevelName définit ou retourne le niveau de date qui s'applique à une mesure cumulée extérieurement.

## Syntaxe

*Column* .DateDegreeofDetailLevelName

## Application

Objet Column

## Explication

Si vous conservez des données structurelles et des données transactionnelles dans des fichiers séparés, vous pouvez spécifier des valeurs de mesure cumulée dans les fichiers de la source de données transactionnelles.

Si un objet Column représente une mesure cumulée de manière externe, utilisez cette propriété pour spécifier le niveau de date auquel Transformer fournit des catégories. Vous devez spécifier un niveau de détail qui correspond à un niveau de la dimension de date.

Pour utiliser cette propriété, vous devez aussi affecter la valeur True à la propriété External de l'objet DataSource.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
.Columns("EuroDate").DateDegreeofDetail = xtrDateLevel.trDateLevelMonth
```

---

## Propriété DateFormat

La propriété DateFormat définit ou retourne le code de format que Transformer utilise pour afficher les dates dans le niveau.

## Syntaxe

*DateLevel* .DateFormat

## Application

Objet DateLevel

## Explication

Pour les niveaux d'une dimension de temps, Transformer utilise des codes avec des parties de date (telles que l'année, le mois, le trimestre et le jour) pour définir la manière dont les formats de date sont affichés. Vous pouvez combiner ces codes pour créer des formats de date personnalisés.

La combinaison des codes vous permet de créer des formats de date tels que :

- "AAAA MMM JJ", qui apparaît sous la forme 1999 Jan 01.
- "AA""T""T", qui apparaît sous la forme 99 Tr1

Code	Description	Exemple (en anglais)
AA	Année représentée avec deux chiffres	99
AAAA	Année représentée avec quatre chiffres	1999
Q	Trimestre représenté avec un chiffre	1
MM	Mois à deux chiffres dans l'année	01
MMM	Nom abrégé du mois	Jan

Code	Description	Exemple (en anglais)
MMMM	Nom complet du mois	Janvier
JJ	Jour représenté avec deux chiffres.	01
JJJJ	Jour de la semaine	Dimanche
/, -, ou caractère espace	Caractères de séparation pour autres codes de date	1999/01/01, 1999-01-01
Toute chaîne entre guillemets	La chaîne	"AJ" affiche l'abréviation AJ

En ce qui concerne les années lunaires, les trimestres portent les libellés T1-4, les mois 1-12/13 et les jours 1-28.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objDateLevel = objDateDim.DimensionLevels("Month")objDateLevel.DateFormat
= "MMMM, yyyy"
```

---

## Propriété DateFunction

La propriété DateFunction définit ou retourne les catégories de date qui sont générées dans un niveau.

## Syntaxe

*DateLevel* .DateFunction

## Application

Objet DateLevel

## Explication

Utilisez cette propriété pour créer des catégories de date appropriées à partir d'une colonne source contenant des dates. Spécifiez, par exemple, la constante trSpecialFunctionMonth pour générer des catégories de mois.

La propriété DateFunction utilise les valeurs de la constante xtrSpecialFunction.

## Type

Constante - xtrSpecialFunction

## Accès

Lecture/écriture

## Exemples

```
objDateWizard = objModel.DateWizardobjDateDim  
= objDateWizard.CreateDateDimension()objDateLevel  
= objDateDim.DimensionLevels(1)objDateLevel.DateFunction  
= xtrSpecialFunction.trSpecialFunctionLunarQuarter
```

---

## Propriété DateInputFormat

La propriété DateInputFormat définit ou retourne l'ordre des éléments année, mois et jour des valeurs d'entrée de date.

### Syntaxe

*Column* .DateInputFormat

### Application

Objet Column

### Explication

Cette propriété utilise les valeurs de la constante xtrDateFormat. Utilisez la constante trPredefined quand le format est défini dans votre source de données, notamment dans une feuille de calcul qui possède des cellules au format date.

### Type

Constante - xtrDateFormat

### Accès

Lecture/écriture

### Exemples

```
objDataSource  
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objDataSource  
.Columns("EuroDate").DateInputFormat  
= xtrDateFormat.trMDY
```

---

## Propriété DateLevel

La propriété DateLevel définit ou retourne l'objet DateLevel associé à un enregistrement de devise.

### Syntaxe

*CurrencyRecord* .DateLevel

### Application

Objet CurrencyRecord

## Explication

Utilisez cette propriété pour spécifier le niveau de la dimension de temps auquel s'appliquent les taux de l'enregistrement de devise, notamment le mois, la semaine ou le jour. Transformer crée une collection d'objets CurrencyRate basée sur le niveau de date spécifié et la plage de dates disponible.

Pour éditer les taux, utilisez la propriété Rate.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - DateLevel

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add(objCurrencyRecord.DateLevel = objLevel
```

---

## Propriété DateWizard

La propriété DateWizard renvoie l'objet DateWizard.

## Syntaxe

*Model* .DateWizard

## Application

Objet Model

## Explication

Utilisez l'objet DateWizard pour simplifier la création d'une dimension de temps et des niveaux de date. Utilisez ensuite la méthode CreateDateDimension pour terminer le processus. Cette procédure ajoute automatiquement un ensemble standard de catégories spéciales (date relative), telles que l'année à ce jour.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - DateWizard

## Accès

Lecture

## Exemples

```
objDateWizard = objModel.DateWizard
```

---

## Propriété DecimalPoint

La propriété DecimalPoint définit ou retourne le caractère de séparation utilisé comme virgule décimale.

### Syntaxe

*Object* .DecimalPoint

### Application

Objet CrossTabDataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

### Explication

Vous ne pouvez pas utiliser un caractère nul (c'est-à-dire, aucun caractère).

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objDataSource.DecimalPoint = "."
```

---

## Propriété Decimals

La propriété Decimals retourne le nombre de décimales défini dans les métadonnées de la source.

### Syntaxe

*Column* .Decimals

### Application

Objet Column

### Explication

Si une source de données, telle qu'un fichier à plat, ne contient pas ces informations relatives aux métadonnées, utilisez les propriétés InputScale et

OutputScale pour mettre à l'échelle les valeurs que PowerPlay affiche à l'exécution.

### **Type**

Long

### **Accès**

Lecture

---

## **Propriété DefaultCategoryOrderBy**

La propriété DefaultCategoryOrderBy définit l'ordre de tri par défaut (croissant) pour toutes les catégories du modèle.

### **Syntaxe**

*Model* .DefaultCategoryOrderBy

### **Application**

Objet Model

### **Discussion**

Utilisez cette propriété pour spécifier que le modèle utilise le paramètre Preference pour trier toutes les catégories du modèle ; cela signifie que les libellés sont triés dans l'ordre alphanumérique croissant.

Cette propriété utilise la valeur trUsePreference.

### **Type**

Constante - xtrDefaultOrderBy

### **Accès**

Lecture/écriture

---

## **Propriété DateFormat**

La propriété DateFormat définit ou retourne le paramètre par défaut de la propriété DateInputFormat.

### **Syntaxe**

*Application* .DateFormat

### **Application**

Objet Application

## Explication

Cette propriété permet d'indiquer que la propriété `DateInputFormat` utilise le format de date prédéfini dans la source de données ou celui défini dans les paramètres régionaux du Panneau de configuration de Windows.

Lorsque la source de données ne définit pas le format de date, tel que dans les fichiers texte, Transformer utilise le format AAAAMMJJ.

Cette propriété utilise les valeurs de la constante `xtrPreferences`.

## Type

Constante - `xtrPreferences`

## Accès

Lecture/écriture

## Exemples

```
objTransApp.DefaultDateFormat = xtrPreferences.trDateFormatFromControlPanel
```

---

## Propriété Description

La propriété `Description` définit ou renvoie une description de l'objet.

## Syntaxe

*object* .**Description**

## Application

Objet Association

Objet CalculationDefinition

Objet Category

Objet CategorySet

Objet ChildCube

Objet Column

Objet CrossTabDataSource

Objet Cube

Objet CubeGroup

Objet CustomView

Objet DataSource

Objet DateDimension



Objet DateLevel  
Objet DbDataSource  
Objet Dimension  
Objet DrillThroughTarget  
Objet FlatFileDataSource  
Objet IqdDataSource  
Objet de niveau  
Objet Measure  
Objet Model  
Objet de code d'accès  
Objet SpecialCategory

### **Explication**

Utilisez cette propriété comme documentation interne, ainsi que pour fournir des informations aux utilisateurs via la fenêtre Explication de PowerPlay.

Par exemple, quand PowerPlay récupère la description d'une mesure calculée, le texte de la fenêtre Explication peut décrire la manière dont la formule du calcul est dérivée.

### **Type**

Chaîne

### **Accès**

Lecture/écriture

### **Exemples**

```
objMeasure.Description = "Forcasted volume for product  
line."
```

---

## **Propriété DesiredPartitionSize**

La propriété DesiredPartitionSize définit ou retourne la taille de partition de préférence pour chaque partition d'un cube.

### **Syntaxe**

*object* .DesiredPartitionSize

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété avec le processus de découpage automatique en partitions pour définir la taille maximum de préférence de chaque partition du cube. La valeur spécifiée limite avec précision la taille de la partition que PowerPlay recherche pour trouver des valeurs quand l'utilisateur passe à un niveau inférieur ou supérieur. Une taille de partition réduite accélère l'accès aux récapitulatifs du niveau supérieur des catégories affectées dans un cube.

Pour définir cette propriété, la propriété Optimize doit spécifier la constante trOptimizeDefault ou trOptimizeAutoPartition. Vous devez également définir les propriétés du cube : EstimatedRows et MaxNumPartLevels.

Pour créer manuellement des partitions, utilisez les propriétés Partition et MaxNumPartLevels.

**Valeur par défaut** : 5 % du nombre estimé d'enregistrements consolidés.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.DesiredPartitionSize  
= 500000
```

---

## Propriété DetachDataSource

La propriété DetachDataSource définit ou retourne une valeur indiquant si la connexion à une source de données est maintenue ou libérée.

## Syntaxe

*Application* .DetachDataSource

## Application

Objet Application

## Explication

Quand cette propriété a la valeur True, Transformer libère une source de données quand il a terminé de remplir un modèle et de créer des cubes, puis se reconnecte à la source de données au moment requis. Si la valeur est False, Transformer demeure connecté à la source de données.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objTransApp.DetachDataSource = True
```

---

## Propriété DetailLevel

La propriété DetailLevel définit ou retourne un objet de niveau qui spécifie le niveau de détail le plus bas des cubes dans un groupe de cubes.

## Syntaxe

*CubeGroup* .DetailLevel

## Application

Objet CubeGroup

## Explication

Quand cette propriété n'est pas définie, Transformer inclut la catégorie à partir du niveau spécifié par la propriété GroupLevel ainsi que toutes les catégories descendantes associées.

Les noms en double sont qualifiés par le nom de la catégorie parent.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion  
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion.DetailLevel  
= objRegionsDrill.Levels("Branch")
```

---

## Propriété Dimension

La propriété Dimension renvoie une dimension pour un objet Category ou SpecialCategory.

### Syntaxe

*Object* .Dimension

### Application

Objet Category

Objet SpecialCategory

### Explication

Les catégories ou catégories spéciales n'appartiennent qu'à une seule dimension.

### Type

Objet - Dimension

### Accès

Lecture

---

## Propriété DimensionInclude

La propriété DimensionInclude définit ou retourne le type de vue d'une vue personnalisée.

### Syntaxe

*CustomView* .DimensionInclude(Dimension)

### Application

Objet CustomView

### Explication

Chaque objet CustomView intègre automatiquement une collection d'objets View, à savoir un pour chaque dimension. La vue par défaut de chaque objet d'une collection est la vue "Toutes catégories". Cette propriété permet d'omettre toutes les catégories ou de créer une vue personnalisée.

Quand vous la définissez, vous devez inclure un objet Dimension comme paramètre. Ce paramètre spécifie la dimension à laquelle s'applique la vue.

Pour définir une vue de façon à ce que toutes les catégories d'une dimension soient omises à partir d'un cube, affectez à cette propriété la constante `trViewTypeOmitDimension`.

Les vues personnalisées fournissent aux utilisateurs de PowerPlay un sous-ensemble spécifique d'informations relatives au cube. Une vue personnalisée peut résumer un niveau entier ou des catégories individuelles du niveau. Si vous spécifiez une vue personnalisée (`trViewTypeCustom`), vous devez aussi utiliser la propriété `DimensionView` pour retourner l'objet `View` associé à une dimension. Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété `Apex` ou associer l'objet `View` aux catégories ou aux niveaux sélectionnés au moyen de la méthode `SetViewStatus`.

Pour terminer une vue personnalisée, vous devez associer une vue personnalisée à un cube. Utilisez la méthode `Add` pour ajouter un objet `CustomView` à la collection `CubeCustomViews` du cube.

La propriété `DimensionInclude` retourne une constante de la liste des valeurs de `xtrViewType`.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Dimension	Requise. Spécifie la dimension à laquelle s'applique la vue.  Type : Dimension

## Type

Constante - `xtrViewType`

## Accès

Lecture/écriture

## Exemples

```
custom_view = model.CustomViews.Add()custom_view.DimensionInclude(dimension) =  
xtrViewType.trViewTypeCustom
```

---

## Propriété `DimensionLevels`

La propriété `DimensionLevels` renvoie une collection `DimensionLevels`.

## Syntaxe

*object* .`DimensionLevels`

## Application

Objet DateDimension

Objet Dimension

## Explication

La collection Levels contient une liste des niveaux uniques de la dimension. Lorsqu'une dimension possède des branches de niveaux secondaires, un niveau individuel peut être inclus plus d'une fois dans la dimension; cette collection ne contient cependant qu'une seule référence à ce niveau, quelle que soit sa fréquence d'apparition dans la dimension.

Quand un niveau figure dans plus d'une branche de niveaux, utilisez la propriété Context pour spécifier le chemin d'accès.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - DimensionLevels

## Accès

Lecture

## Exemples

```
objLevel = objModel.Dimensions("Products").DimensionLevels(1)
```

---

## Propriété DimensionName

La propriété DimensionName définit le nom d'un nouvel objet DateDimension.

## Syntaxe

*DateWizard* .**DimensionName**

## Application

Objet DateWizard

## Explication

Si vous ne spécifiez pas le nom de la dimension de temps, Transformer génère une erreur.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Ecriture

## Exemples

```
objDateWizard.DimensionName = "Dates"
```

---

## Propriété Dimensions

La propriété Dimensions retourne une collection de tous les objets Dimension et DateDimension dans un modèle.

### Syntaxe

*Model* .Dimensions

### Application

Objet Model

### Discussion

Une fois que la collection est retournée, vous pouvez utiliser la méthode Item pour retourner une dimension spécifique, puis modifier des propriétés.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet - Dimensions

## Accès

Lecture

## Exemples

```
objTimeDimension = objModel.Dimensions.Item("Time")
```

---

## Propriété DimensionView

La propriété DimensionView définit ou retourne l'objet View associé à un cube ou à une vue personnalisée.

### Syntaxe

*object* .DimensionView(Dimension)

### Application

Objet Cube

Objet CubeGroup

Objet CustomView

## Explication

Quand vous la définissez, vous devez inclure un objet Dimension comme paramètre. Ce paramètre spécifie la dimension à laquelle s'applique la vue.

Utilisez la propriété DimensionView quand vous créez des vues personnalisées d'une dimension. Les vues personnalisées fournissent aux utilisateurs de PowerPlay un sous-ensemble spécifique d'informations relatives au cube. Une vue personnalisée peut notamment résumer un niveau entier ou des hiérarchies de catégories individuelles au sein du niveau.

Chaque objet Dimension inclut automatiquement une collection de deux objets View. Par défaut, la propriété DimensionView a pour valeur le premier objet de la collection Views, à savoir : "Toutes les catégories". Pour omettre une dimension à partir d'un cube, affectez à la propriété DimensionView le deuxième objet de la collection, à savoir : "Omettre la dimension". Pour créer une vue personnalisée d'une dimension, utilisez d'abord la méthode Add pour ajouter un objet View à la collection, puis affectez la valeur trViewTypeCustom à la propriété ViewType.

Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété Apex ou associer l'objet View aux catégories ou aux niveaux sélectionnés au moyen de la méthode SetViewStatus. Vous pouvez ensuite utiliser le nouvel objet View pour définir la propriété DimensionView.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Vues personnalisées

Si vous spécifiez une vue personnalisée à l'aide de la propriété DimensionInclude, vous devez utiliser ensuite la propriété DimensionView pour retourner l'objet View associé à une dimension. Vous pouvez ensuite créer une catégorie racine à l'aide de la propriété Apex ou associer l'objet View aux catégories ou aux niveaux sélectionnés au moyen de la méthode SetViewStatus.

Pour terminer une vue personnalisée, vous devez associer une vue personnalisée à un cube. Pour cela, utilisez la méthode Add afin d'ajouter un objet CustomView à la collection CubeCustomViews du cube.

Paramètre	Description
Dimension	Requise. Spécifie la dimension pour laquelle une vue personnalisée est définie.  Type : Objet

## Type

Objet



## Accès

Lecture/Ecriture (Cube)

Lecture (CustomView)

## Exemples

```
objCube = objModel.Cubes("Sales and Marketing")objCube.DimensionView(objDimension)  
= objView
```

---

## Propriété DimensionViewType

La propriété DimensionViewType retourne le type de vue associé à une dimension dans un cube.

## Syntaxe

```
object .DimensionViewType(Dimension [, View] )
```

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété pour déterminer si aucune, plusieurs ou toutes les catégories de la dimension sont incluses.

La propriété DimensionViewType utilise les valeurs de la constante xtrViewType.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
Dimension	Requise. Spécifie une dimension. Type : Objet
Vue	Facultatif. Spécifie une vue de type trViewTypeCustom. Type : Objet

## Type

Constante - xtrViewType

## Accès

Lecture/écriture

## Exemples

```
objCube  
= objModel.Cubes("Sales and Marketing")If objCube.  
  DimensionViewType(objDimension)  
= _xtrViewType.trViewTypeAllCategories Then
```

---

## Propriété DisplayName

La propriété DisplayName définit ou retourne le nom à afficher pour l'objet SecurityObject.

### Syntaxe

*SecurityObject* .DisplayName

### Application

Objet SecurityObject

### Explication

Cette propriété permet de définir le nom d'affichage affiché dans l'interface utilisateur Transformer sous les objets Security.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
new_namespace.DisplayName = Name_of_Object
```

---

## Propriété DrillCode

La propriété DrillCode définit ou retourne un code qui identifie de manière unique la catégorie d'accès au détail dans toute la dimension.

### Syntaxe

*object* .DrillCode

### Application

Objet DateDrillDown

Objet DrillDown

### Explication

Le code, qui peut être un identificateur alphabétique ou alphanumérique (notamment Par gamme), est traité par Transformer comme un type de données chaîne.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objDateDim = objDateWizard.CreateDateDimension()objDrillDown  
= objDateDim.DrillDowns(1)objDrillDown.DrillCode = "By Dates"
```

---

## Propriété DrillDowns

La propriété DrillDowns retourne une collection d'objets DateDrillDown ou DrillDown.

## Syntaxe

*object* .DrillDowns

## Application

Objet DateDimension

Objet Dimension

## Explication

L'objet DateDrillDown et l'objet DrillDown définissent tous deux une branche de niveaux.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - DrillDowns

## Accès

Lecture

## Exemples

```
objDrill = objModel.Dimensions("Retailers").DrillDowns(2)
```

---

## Propriété DrillInclusion

La propriété DrillInclusion définit ou retourne une valeur indiquant si une branche de niveaux est incluse dans un cube.

## Syntaxe

*object* .DrillInclusion

## Application

Objet DateDrillDown

Objet DrillDown

## Explication

Dans PowerPlay, une catégorie de branche incluse apparaît sous forme de dossier contenant les catégories enfants qui sont associées au chemin d'accès de cette branche. Une catégorie de branche supprimée n'apparaît pas dans PowerPlay, à la différence des catégories enfants.

La propriété DrillInclusion n'utilise que trois constantes de la liste des valeurs de xtrInclusion :

- trInclusionDefault (affiche la catégorie de branche),
- trInclusionSuppress (masque la catégorie de branche),
- trInclusionGenerate (affiche la catégorie de branche).

## Type

Constante - xtrInclusion

## Accès

Lecture/écriture

## Exemples

```
objDrillDown = objDateDim.DrillDowns(1)objDrillDown.DrillInclusion  
= xtrInclusion.trInclusionSuppress
```

---

## Propriété DrillThroughTargets

La propriété DrillThroughTargets retourne une collection d'objets cible d'accès au détail associés à une mesure ou un cube (cube enfant ou groupe de cubes).

## Syntaxe

*Object* .DrillThroughTargets

## Application

Objet ChildCube

Objet Cube

Objet CubeGroup

Objet Measure

## Explication

Cette propriété permet d'effectuer une itération dans les cibles d'accès au détail d'un objet Model ou bien encore d'ajouter, de modifier ou de supprimer une cible d'accès au détail.

La propriété DrillThroughTargets peut représenter les entités suivantes :

- PowerCubes (.mdc),
- Rapports PowerPlay (.ppr, .ppx)
- Fichiers Impromptu Web Query et Cognos Query (.iwq, iqd)
- Rapports Impromptu (.imr)
- Sources OLAP tierces locales (fichiers .cub MS SSOS, Partitions liées Essbase ou objets Reporting),
- Fichiers associés (.mac, .doc, .xls, .ppt).

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - DrillThroughTargets

## Accès

Lecture

## Exemples

```
objReport = objMeasure.DrillThroughTargets.Add(strReportPath, "Default Report")
```

---

## Propriété DuplicateRollup

La propriété DuplicateRollup définit ou retourne une valeur indiquant la manière dont les valeurs de mesure des enregistrements consolidés sont cumulées.

## Syntaxe

*Measure* .DuplicateRollup

## Application

Objet Measure

## Explicatiion

Cette propriété permet de spécifier la fonction que Transformer exécute pour récapituler des enregistrements doubles dans les données sources. Quand cette propriété a pour valeur trDuplicateRollupSum, par exemple, le cumul de l'enregistrement consolidé correspond à la somme des valeurs de mesure des enregistrements doubles. Transformer exécute le cumul des doubles avant le cumul ordinaire.

Quand la valeur de cette propriété n'est pas trDuplicateRollupNone, elle a la priorité sur le paramètre trConsolidateNO de la propriété Consolidate et force Transformer à consolider le fichier de la source de données pendant la génération des cubes.

N'oubliez pas de définir la propriété DuplicateWeight quand la propriété DuplicateRollup a pour valeur trDuplicateRollupAverage.

Cette propriété utilise les valeurs de la constante `xtrDuplicateRollup`.

## Type

Constante - `xtrDuplicateRollup`

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure  
= objMeasures("Revenue")currentMeasure.DuplicateRollup  
= xtrDuplicateRollup.trDuplicateRollupAverage
```

---

## Propriété DuplicateWeight

La propriété `DuplicateWeight` définit ou retourne le nom de la mesure qui contient des facteurs de pondération.

## Syntaxe

*Measure* .`DuplicateWeight`

## Application

Objet `Measure`

## Explication

Utilisez cette propriété pour retourner une moyenne pondérée de la mesure de cumul plutôt qu'une moyenne réelle. La moyenne pondérée de la mesure A, qui utilise la mesure B comme pondération, est calculée à l'aide d'une équation similaire à celle-ci :

$$\sum(A_i * B_i) / \sum B_i$$

La mesure qui est cumulée doit être définie à l'aide de la constante `trDuplicateRollupAverage`.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure = objMeasures("Revenue")currentMeasure  
currentMeasure.DuplicateWeight = objModel.Measures("Quantity").Name
```

---

## Propriété EarliestDate

La propriété EarliestDate définit ou retourne la date la plus ancienne dans une plage de catégories de dates.

### Syntaxe

*object* .EarliestDate

### Application

Objet DateDimension

Objet DateWizard

### Explication

Utilisez cette propriété avec la propriété LatestDate pour limiter la plage des catégories de date acceptables qui sont incluses dans une dimension de temps. Vous spécifiez une plage pour éliminer des catégories telles que les années précédentes qui ne sont pas significatives aux yeux des utilisateurs de PowerPlay.

Pour générer des catégories de date uniquement, utilisez la propriété GenerateTimePeriod (objet DateDimension) ou la propriété GenerateDates (objet DateWizard).

Si Transformer rencontre des valeurs de date qui ne figurent pas dans la plage spécifiée, il génère une catégorie Early Dates et/ou Late Dates, selon l'échéance des dates qui ne sont pas comprises dans la plage. Si Transformer ne rencontre aucune date antérieure ou postérieure et si les dates ne peuvent pas être placées dans la plage spécifiée, il génère une catégorie Invalid Dates.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

### Accès

Ecriture (DateWizard)

Lecture/Ecriture (DateDimension)

### Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.EarliestDate  
= 19930101
```

---

## Propriété EMUEntryDate

La propriété EMUEntryDate définit ou retourne la date à laquelle commencent les calculs de triangulation de l'euro d'une devise.

## Syntaxe

*CurrencyRecord* .EMUEntryDate

## Application

Objet CurrencyRecord

## Explication

Cette propriété utilise le format de date AAAAMMJJ. Pour l'employer, vous devez affecter la valeur True à la propriété CurrencyIsEMU.

**Valeur par défaut :** 19990101

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRec = objModel.CurrencyRecords(1)objCurrencyRec.EMUEntryDate  
= "19990101"
```

---

## Propriété EnableMessageLogging

La propriété EnableMessageLogging définit ou retourne une valeur indiquant si les messages de Transformer sont consignés dans un fichier journal.

## Syntaxe

*Application* .EnableMessageLogging

## Application

Objet Application

## Explication

Cette propriété permet de créer un fichier journal. Une fois que vous lui avez affecté la valeur True, vous pouvez utiliser :

- la propriété LogErrorLevel pour spécifier la gravité des messages d'erreur consignés,
- la propriété LogFileName pour spécifier le nom du fichier journal,
- la propriété LogFilesPath pour définir l'emplacement dans lequel les fichiers journaux sont consignés.

**Paramètre par défaut :** True

## Type

Booléen



## Accès

Lecture/écriture

## Exemples

```
objTransApp.EnableMessageLogging = True
```

---

## Propriété EnableTimePeriod

La propriété EnableTimePeriod définit le niveau de détail d'une dimension de temps.

### Syntaxe

```
DateWizard .EnableTimePeriod(TimeType)
```

### Application

Objet DateWizard

### Explication

Cette propriété permet de spécifier les niveaux de date que vous ajoutez à une dimension de temps, notamment l'année, le trimestre et le mois. Si vous n'utilisez pas cette propriété, Transformer base les hypothèses sur la source de données.

Le paramètre TimeType indique le niveau de détail de la date. Si sa valeur est trTimeTypeMonth, par exemple, la période en cours affiche l'année jusqu'à ce mois.

Cette propriété utilise les valeurs de la constante xtrTimeType.

Paramètre	Description
TimeType	Requise. Fournit la valeur de la constante xtrTimeType.  Type : Constante

### Type

Booléen

### Accès

Écriture

### Exemples

```
objDateWizard  
= objModel.DateWizardobjDateWizard.EnableTimePeriod(xtrTimeType.trTimeTypeYear)  
= True
```

---

## Propriété EstimatedRows

La propriété EstimatedRows définit ou retourne une estimation du nombre d'enregistrements contenus dans le cube.

## Syntaxe

*object* .EstimatedRows

## Application

Objet Cube

Objet CubeGroup

## Explication

Transformer utilise l'estimation fournie, puis met à jour la valeur du processus de génération du cube pour refléter le nombre réel d'enregistrements consolidés qui sont ajoutés au cube.

Pour définir cette propriété, la propriété Optimize doit spécifier la constante trOptimizeDefault ou trOptimizeAutoPartition. Vous devez également définir les propriétés du cube : DesiredPartitionSize et MaxNumPartLevels.

Pour créer manuellement des partitions, utilisez les propriétés Partition et MaxNumPartLevels.

**Valeur par défaut :** 10 000 000

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.EstimatedRows  
= 10000000
```

---

## Propriété ExcludeAutoPartition

La propriété ExcludeAutoPartition définit ou retourne une valeur indiquant si une dimension spécifiée est exclue du processus de découpage automatique en partitions.

## Syntaxe

*object* .ExcludeAutoPartition

## Application

Objet DateDimension

Objet Dimension

## Explication

Quand cette propriété a la valeur True, aucun niveau de partition n'est créé pour la dimension lors de la création du cube.

Les partitions résument les données du cube dans plusieurs partitions subordonnées pour accélérer la récupération dans PowerPlay. Comme le découpage en partitions peut augmenter la durée requise pour créer le cube dans Transformer, utilisez la propriété ExcludeAutoPartition afin d'exclure une dimension qui ne requiert pas ce type d'optimisation.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDimensions = objModel.DimensionsobjLocationsDim  
= objDimensions.Item(3)objLocationsDim.ExcludeAutoPartition  
= True
```

---

## Propriété ExpressionText

La propriété ExpressionText définit ou retourne une expression utilisée dans une opération mathématique ou une opération de chaîne. Dans l'édition en cours, les expressions conditionnelles if-then-else sont désormais prises en charge.

## Syntaxe

*object* .ExpressionText

## Application

Objet CalculationDefinition

Objet Category

Objet Column

Objet Measure

Objet SpecialCategory

## Explication

Cette propriété permet de spécifier les éléments utilisés pour créer une expression. Une expression peut être constituée d'une fonction et d'opérateurs mathématiques et fait généralement référence à des mesures, des colonnes ou des catégories comme source des valeurs.

Les expressions peuvent comporter du texte, des dates ou des données numériques, selon le type de données de l'objet référencé. Transformer évalue les expressions numériques suivant des règles mathématiques standard.

Quand vous référencez une colonne dans la propriété ExpressionText, sa propriété DataClass doit avoir une valeur d'entrée valide sous peine de générer une erreur d'exécution. Les paramètres non valides sont trDataClassIgnore et trDataClassDefault, où la valeur par défaut retourne des résultats dans une classe de données non spécifiée.

Quand vous avez terminé de spécifier l'expression, vous pouvez utiliser la propriété IsExpressionValid pour déterminer si tous ses composants sont valides.

Lorsqu'elle est utilisée avec un objet Measure, la propriété ExpressionText définit une mesure en tant que mesure calculée.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
calcDef = dimension.CalculationDefinitions.Add()calcDef.ExpressionText
= "share ( catset ( ""Set 1"" ) , "" & _ parentCategory.Code
& """)"
```

---

## Propriété External

La propriété External définit ou retourne une valeur indiquant si la source de données contient des valeurs récapitulatives.

## Syntaxe

*object* .External

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

Si vous conservez des données structurées et des données transactionnelles dans des fichiers séparés, vous pouvez récapituler préalablement des valeurs de mesure dans les fichiers de la source de données transactionnelles.

Utilisez les sources de données structurées pour définir vos dimensions. Pour chaque source de données transactionnelles, affectez la valeur `True` à la propriété `External`, puis ajoutez les mesures à la collection `Measures`. Affectez la constante `trRollupExternal` à la propriété `RegularRollup` de chaque objet `Measure`. Utilisez ensuite la méthode `AssociateWith` de chaque dimension applicable pour spécifier une colonne à partir d'une source de données transactionnelles. La colonne doit correspondre à une colonne de la source de données structurées.

Si le modèle alloue des mesures, utilisez la propriété `DateDegreeofDetailLevelName` pour spécifier le degré de détail.

Paramètre par défaut : `False`

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objDataSource.External
= False
```

---

## Propriété `FieldSeparator`

La propriété `FieldSeparator` définit ou retourne le type de séparateur de zones utilisé dans une source de données.

## Syntaxe

*FlatFileDataSource* .**FieldSeparator**

## Application

Objet `FlatFileDataSource`

## Explication

Cette propriété permet de spécifier le caractère servant de séparateur de zones dans une source de données à zones délimitées. Transformer analyse seulement le premier octet des caractères à deux et à plusieurs octets pour déterminer si un caractère du flux d'entrée correspond au caractère de séparation spécifié.

Cette propriété ne s'applique qu'aux fichiers à plat dont la propriété `SourceType` a pour valeur la constante `trFlatFile` ou `trFlatFileColumnNames`.

Les séparateurs standard, tels la virgule, le point virgule ou le caractère espace, restent les mêmes, quel que soit le jeu de caractères. Si les données sources sont rendues dans le jeu de caractères de la page de codes DOS (OEM), Transformer remplace le caractère de séparation par le caractère du jeu OEM.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources.
  Add(xtrObjectType.trFlatFileDataSource)objDataSource.FieldSeparator
= ", "
```

---

## Propriété FileName

La propriété FileName renvoie le nom d'un fichier modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.

## Syntaxe

*Model* .FileName

## Application

Objet Model

## Discussion

Utilisez cette propriété pour afficher le nom d'un fichier modèle ou pour consigner le nom dans un fichier journal.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

## Exemples

```
MsgBox(objModel.FileName & " " & .Size &
" " & .Time)
```

---

## Propriété Filters

La propriété Filters renvoie la collection d'objets Filter associés à une requête.

## Syntaxe

*Query* .Filters

## Application

Objet Query

## Explication

Cette propriété permet d'effectuer une itération dans les filtres d'un objet Query, ainsi que d'ajouter, de modifier et de supprimer un filtre.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Filtre

## Accès

Lecture

## Exemples

```
new_package = model.Packages.Add()new_query = new_package.Queries.Add()new_filter  
= new_query.Filters.Add()
```

---

## Propriété FindCategoryByCatCode

La propriété FindCategoryByCatCode retourne l'objet Category contenant la chaîne du code de catégorie spécifiée.

## Syntaxe

*Dimension* .FindCategoryByCatCode

## Application

Objet Dimension

## Explication

Paramètre	Description
CategoryCode	Requise. Nom du code de catégorie. Type : chaîne

## Type

Objet - Category

## Accès

Lecture

---

## Propriété Format

La propriété Format définit et renvoie le format pour des valeurs numériques.

### Syntaxe

*object* .Format

### Application

Objet Category

Objet Measure

Objet SpecialCategory

### Explication

Cette propriété permet de spécifier le format des valeurs numériques. PowerPlay utilise les valeurs formatées dans les rapports PowerPlay. Vous devez placer la chaîne du format entre guillemets ("").

Pour ajouter des décimales à une valeur de mesure, utilisez la propriété FormatDecimals.

Les formats disponibles sont énumérés ci-dessous. Plusieurs fonctions de format, telles que le symbole monétaire et le séparateur des milliers, peuvent changer en fonction des propriétés que vous définissez dans les paramètres régionaux du Panneau de configuration de Windows.

Format	Exemple
0	1000000
#,##0	1,000,000
\$0	\$1000000
\$#,##0	\$1,000,000
0%	100%
%0	%100
0E+00	1E+09
0K	1000K
# ##0K	1,000K
K0	K1000
K#,##0	K1,000



Format	Exemple
\$0K	\$1000K
\$#,##0K	\$10 000K
0M	1000M
#,##0M	1,000M
M0	M1000
M#,##0	M1,000
\$0M	\$1000M
\$#,##0M	\$1,000M

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objMeasures
= objModel.MeasurescurrentMeasure = objMeasures("Revenue").currentMeasure.Format
= "$#,##0"
```

---

## Propriété FormatDecimals

La propriété FormatDecimals définit ou retourne le nombre de décimales que PowerPlay affiche pour une mesure ou des catégories calculées.

## Syntaxe

*Object* .FormatDecimals

## Application

Objet Category

Objet Measure

## Explication

Utilisez cette propriété comme paramètre de format.

Plage : 0 à 9.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long (objet Measure), Int (objet Category)

### Accès

Lecture/écriture

### Exemples

```
objMeasures = objModel.MeasurecurrentMeasure  
= objMeasures("Revenue").currentMeasure.FormatDecimals  
= 2
```

---

## Propriété FullName

La propriété FullName renvoie l'emplacement d'un fichier modèle.

### Syntaxe

*Model* .FullName

### Application

Objet Model

### Explication

Utilisez cette propriété pour afficher le nom et l'emplacement d'un fichier modèle ou pour consigner le nom de fichier complet dans un fichier journal.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Chaîne

### Accès

Lecture

---

## Propriété GenerateCategories

La propriété GenerateCategories définit ou retourne une valeur indiquant si des catégories sont générées pour la source de données.

### Syntaxe

*object* .GenerateCategories

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

Objet Query

## Explication

Quand cette propriété a pour valeur False, Transformer ignore la source de données chaque fois que des catégories sont générées. Affectez la valeur False à cette propriété dans les cas suivants :

- quand vous possédez plusieurs sources de données et que vous souhaitez exclure la génération de catégories pour l'une d'elles,
- pendant la mise à jour incrémentielle d'autres sources de données,
- quand vous possédez une source de données de taux de conversion pour laquelle vous ne devez pas générer des catégories de date.

Vous pouvez aussi utiliser la propriété GeneratePowerCube pour contrôler la manière dont Transformer référence des sources de données.

Valeur par défaut : True

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDataSource  
= objModel.DataSources.  
  Add(xtrObjectType.trFlatFileDataSource)objDataSource.GenerateCategories  
= True
```

---

## Propriété GenerateDateCategories

La propriété GenerateDateCategories définit ou retourne une valeur indiquant si un niveau de date génère des catégories de date.

## Syntaxe

*DateLevel*.GenerateDateCategories

## Application

Objet DateLevel

## Explication

Pour générer des catégories pour un niveau de date, affectez à cette propriété la constante trGenerateDatesAll. Pour supprimer la génération de catégories de date pour un niveau, affectez à cette propriété la constante trGenerateDatesNone.

Cette propriété utilise les valeurs de la constante xtrDateCategoriesGeneration.

## Type

Constante - xtrDateCategoriesGeneration

## Accès

Lecture/écriture

## Exemples

```
objDateDim = objModel.Dimensions("Date")objDateDim.GenerateDateCategories  
= xtrDateCategoriesGeneration.trGenerateDatesAll
```

---

## Propriété GenerateDates

La propriété GenerateDates définit si l'objet DateWizard génère des catégories de dates.

## Syntaxe

```
DateWizard .GenerateDates
```

## Application

Objet DateWizard

## Explication

Cette propriété permet de générer des catégories de dates.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Booléen

## Accès

Ecriture

---

## Propriété GeneratePowerCube

La propriété GeneratePowerCube définit ou retourne une valeur indiquant à quel moment une source de données est référencée par un modèle.

### Syntaxe

*object* .GeneratePowerCube

### Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

Objet Query

### Explication

Cette propriété vous permet de déterminer si une source de données spécifique est référencée pendant la génération de catégorie et/ou la création de cube. Dans les modèles utilisant plusieurs sources de données, vous pouvez limiter l'accès à certaines sources de données pour éviter tout traitement superflu ainsi que pour optimiser l'efficacité du modèle.

La propriété GeneratePowerCube utilise les valeurs de la constante `xtrPowerCubeGeneration`.

Vous pouvez aussi utiliser la propriété `GenerateCategories` pour contrôler les sources de données référencées par `Transformer` pendant la création de catégorie.

### Type

Constante - `xtrPowerCubeGeneration`

### Accès

Lecture/écriture

### Exemples

```
objDataSource
= objModel.DataSources.Add(xtrObjectType
    .trFlatFileDataSource)objDataSource.GeneratePowerCube
= xtrPowerCubeGeneration.trGenerationDefault
```

---

## Propriété GenerateTimePeriod

La propriété GenerateTimePeriod définit ou retourne les options de génération de catégorie pour une dimension de temps.

## Syntaxe

*DateDimension* .**GenerateTimePeriod**(TimeType)

## Application

Objet DateDimension

## Explication

Utilisez cette propriété pour spécifier le degré de détail des dates avant de générer des catégories pour une dimension de temps. Le paramètre TimeType définit le niveau auquel Transformer autorise la génération de rapports à partir des colonnes relatives aux dates dans une source de données. Ce niveau peut être l'année, le trimestre, le mois, la semaine ou le jour.

Utilisez les propriétés EarliestDate et LatestDate pour définir la plage de catégories de date générées.

Le paramètre TimeType doit correspondre à une des constantes de xtrTimeType. La propriété définit ou retourne une des constantes de xtrGenerateOptions.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
TimeType	Requise. Spécifie la valeur de la constante xtrTimeType.  Type : Constante

## Type

Constante - xtrTimeType

## Accès

Lecture/écriture

## Exemples

```
objDateDim  
= objModel.Dimensions("Date")objDateDim.  
GenerateTimePeriod(xtrTimeType.trTimeTypeMonth)  
= _ xtrGenerateOptions.trGenerateAll
```

---

## Propriété Group

La propriété Group définit ou retourne une valeur indiquant si des catégories calculées sont regroupées.

## Syntaxe

*CalculationDefinition* .**Group**

## Application

Objet CalculationDefinition

## Explication

Quand cette propriété a la valeur True, les catégories calculées à chaque niveau sont regroupées dans un rapport PowerPlay. Transformer regroupe des catégories calculées quand l'expression s'applique à toutes les catégories d'un niveau.

Quand cette propriété a la valeur False, chaque catégorie calculée du rapport est adjacente à la catégorie qu'elle référence.

Paramètre par défaut : True

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
calcDef = dimension.CalculationDefinitions.Add()calcDef.Group  
= False
```

---

## Propriété GroupDimension

La propriété GroupDimension définit ou retourne la dimension utilisée pour créer un groupe de cubes.

## Syntaxe

*CubeGroup*.GroupDimension

## Application

Objet CubeGroup

## Explication

Un groupe de cubes référence une dimension unique. Utilisez cette propriété pour spécifier cette dimension, puis la propriété GroupLevel pour définir les cubes enfants, à savoir un pour chaque catégorie du niveau indiqué.

Vous utilisez ensuite les propriétés DetailLevel et SummaryLevel pour définir le degré de détail afin que les utilisateurs de PowerPlay ne puissent voir que les données les plus significatives.

Par exemple, votre société possède quatre services de ventes régionaux. Vous créez un groupe de cubes pour le directeur de chaque service afin qu'il puisse effectuer un suivi des résultats des ventes de son service pour le dernier trimestre écoulé.

Puisqu'il n'est pas nécessaire que les directeurs voient les détails des trois autres services, vous définissez le groupe de manière à ne montrer qu'un récapitulatif des résultats des autres services.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion.GroupDimension
= objModel.Dimensions("Sales region")
```

---

## Propriété GroupLevel

La propriété GroupLevel définit ou retourne le niveau utilisé pour définir les cubes individuels d'un groupe de cubes.

## Syntaxe

*CubeGroup*.GroupLevel

## Application

Objet CubeGroup

## explication

Un groupe de cubes référence une dimension unique. Utilisez la propriété GroupDimension pour spécifier cette dimension, puis la propriété GroupLevel pour définir les cubes enfants, à savoir un pour chaque catégorie du niveau indiqué.

Vous utilisez ensuite les propriétés DetailLevel et SummaryLevel pour définir le degré de détail afin que les utilisateurs de PowerPlay ne puissent voir que les données les plus significatives.

Par exemple, votre société possède quatre services de ventes régionaux. Vous créez un groupe de cubes pour le directeur de chaque service afin qu'il puisse effectuer un suivi des résultats des ventes de son service pour le dernier trimestre écoulé. Puisqu'il n'est pas nécessaire que les directeurs voient les détails des trois autres services, vous définissez le groupe de manière à ne montrer qu'un récapitulatif des résultats des autres services.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion.GroupLevel
= objRegionsDrill.Levels("Sales region")
```

---

## Propriété HasSubdimension

La propriété HasSubdimension renvoie une valeur indiquant si un niveau contient une sous-dimension.

## Syntaxe

*object* .HasSubdimension

## Application

Objet DateLevel

Objet de niveau

## Explication

Les sous-dimensions fournissent différents niveaux de détail pour des catégories spécifiques, aussi connues sous le nom de hiérarchies non équilibrées. Les catégories d'une sous-dimension sont indépendantes des niveaux des autres parties de la dimension. Les modifications apportées aux niveaux extérieurs à la sous-dimension n'affectent pas les catégories d'une sous-dimension. L'inverse est également vrai.

Certaines filiales peuvent rapporter les ventes de produit jusqu'au niveau de l'article tandis que d'autres peuvent se limiter au niveau du produit. Vous pouvez avoir une sous-dimension pour les branches qui rapportent jusqu'au niveau de l'article.

## Type

Booléen

## Accès

Lecture

---

## Propriété HideValue

La propriété HideValue spécifie si la valeur d'un objet Category doit être masquée. Lorsque la propriété a la valeur True, la valeur associée à cette catégorie est masquée.

Il est conseillé de définir cette option pour le parent de catégories d'une dimension de scénario, conjointement à l'utilisation de la méthode `SetDefaultCategory`. Cette propriété est appliquée de la même manière par le biais d'OLE qu'avec une autre interface utilisateur.

Valeur par défaut : False

### **Syntaxe**

*Category* .HideValue

### **Application**

Objet Category

### **Accès**

Lecture/écriture

---

## **Propriété ID**

La propriété ID définit ou renvoie l'ID d'espace-noms.

### **Syntaxe**

*Namespace* .ID

### **Application**

Objet Namespace

### **Explication**

Cette propriété permet de définir l'ID d'espace-noms.

### **Type**

Chaîne

### **Accès**

Lecture/écriture

### **Exemples**

```
new_namespace.ID = ID_of_Namespace
```

---

## **Propriété IgnoreMissingValue**

La propriété `IgnoreMissingValue` dans Transformer Series 7 Version 4 et dans les éditions ultérieures, spécifie si les valeurs nulles ou manquantes de type Moyenne ou Moyenne pondérée du cumul d'état temporel doivent être ignorées. Lorsque la propriété est définie sur True, la valeur associée à cette mesure n'est pas incluse dans le calcul de cumul.

Valeur par défaut : False

**Remarque :** cette propriété ne peut pas être définie pour des cumuls d'état temporel de type Première période, Dernière période ou Période en cours. Dans ces cas-là; les valeurs nulles et manquantes sont toujours incluses dans le calcul de cumul. Les valeurs de données manquantes (null) sont toujours exclues des calculs de cumul Min et Max, qu'elle soient définies par Transformer pour afficher "0" ou "n/d" (paramètre ND). Cette propriété est appliquée de la même manière par le biais d'OLE qu'avec une autre interface utilisateur.

## Syntaxe

*Measure* .IgnoreMissingValue

## Application

Objet Measure

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure
= objMeasures("Revenue")currentMeasure.IgnoreMissingValue
= False
```

---

## Propriété Inclusion

La propriété Inclusion définit ou retourne une valeur indiquant à quel moment une catégorie est incluse dans un cube.

## Syntaxe

*object* .Inclusion

## Application

Objet Category

Objet DateLevel

Objet de niveau

Objet SpecialCategory

## Explication

Cette propriété permet de spécifier les catégories à inclure dans un cube. Vous pouvez, par exemple, exclure des catégories dont la valeur source est vide.

Quand vous définissez la propriété Inclusion d'un objet de niveau, elle s'applique à toutes les catégories du niveau ayant pour valeur trInclusionDefault.

Quand vous définissez la propriété Inclusion d'un objet Category, elle s'applique à cette catégorie seulement.

La propriété Inclusion utilise les valeurs de la constante xtrInclusion.

## Type

Constante - xtrInclusion

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim
= objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.Inclusion
= xtrInclusion.trInclusionGenerate
```

---

## Propriété IncrementalUpdate

La propriété IncrementalUpdate définit ou retourne une valeur indiquant si la mise à jour incrémentielle d'un cube ou d'un groupe de cubes est appliquée à partir d'une source de données.

## Syntaxe

*object* .IncrementalUpdate

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété pour mettre à jour des cubes qui sont associés à des sources de données contenant des données incrémentielles. Les données antérieures du cube sont conservées.

Nous vous conseillons de recréer la totalité du cube à intervalles réguliers pour optimiser le plan de découpage en partitions. Vous souhaitez peut-être mettre le cube à jour quotidiennement et le recréer chaque semaine. Chaque fois que vous apportez des modifications structurelles à votre modèle, vous devez recréer le cube avant d'effectuer une autre mise à jour incrémentielle.

Quand cette propriété a la valeur True pour un groupe de cubes, tous les cubes au sein du groupe sont mis à jour de manière incrémentielle.

**Paramètre par défaut :** False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.IncrementalUpdate  
= False
```

---

## Propriété InputScale

La propriété InputScale définit ou retourne la valeur d'échelle utilisée pour convertir les nombres décimaux des colonnes en entiers.

## Syntaxe

*Column* .InputScale

## Application

Objet Column

## Explication

Utilisez cette propriété avec les types de données autres que Float pour convertir les valeurs décimales en valeurs d'entier. La valeur spécifiée définit comment les valeurs sources sont multipliées lors de leur lecture à partir de votre source de données.

Si cette propriété a la valeur 2, par exemple, Transformer écrit la valeur source 94,5 dans le cube sous la forme 9450 ; c'est-à-dire  $94,5 \times 10^2$ . PowerPlay affiche ensuite la valeur 9450.

Utilisez les propriétés OutputScale et FormatDecimals pour rétablir la valeur d'origine des valeurs mises à l'échelle. La propriété OutputScale spécifie la manière dont les valeurs source sont divisées pendant l'exécution de PowerPlay. En résumé, la propriété InputScale supprime les décimales pendant les calculs, tandis que les propriétés OutputScale et FormatDecimals les replacent.

Si OutputScale a pour valeur 2, par exemple, PowerPlay affiche la valeur 95 (arrondi). Si FormatDecimals a pour valeur 1, PowerPlay affiche la valeur 94,5.

La plage des valeurs d'échelle est comprise entre -16 et 16, ce qui correspond à  $10^{-16}$  à  $10^{16}$ .

### Valeur par défaut : 0

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur. Par exemple, une exception se produit avec Update() si la valeur est en dehors de la plage allant de -16 à 16.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objDataSource  
= objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)objColumn.InputScale  
= 0
```

---

## Propriété IsAnyColumnMismatched

La propriété sAnyColumnMismatched retourne une valeur indiquant si des colonnes de la source de données correspondent à des données sous-jacentes.

## Syntaxe

*object* .IsAnyColumnMismatched

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

Objet Query

## Explication

Utilisez cette propriété pour vérifier si les colonnes définies dans le modèle concordent avec celles des fichiers sources. La propriété retourne la valeur False s'il n'y a pas de non-concordance et True s'il y en a une.

Lorsque vous créez un modèle, les colonnes de chaque source de données sont enregistrées dans la définition de modèle. Si vous reclassez, ajoutez, supprimez ou renommez les colonnes des fichiers sources, vous devez mettre le modèle à jour.

## Type

Booléen

## Accès

Lecture

## Exemples

```
new_package = model.Packages.Add()new_query  
= new_package.Queries.Add()If new_query.IsAnyColumnMismatched  
= True Then
```

---

## Propriété IsBad

La propriété IsBad retourne une valeur indiquant si le modèle en attente est corrompu ou récupérable.

### Syntaxe

*SuspendedModel* .IsBad

### Application

Objet SuspendedModel

### Explication

Un modèle en attente est créé quand il ne se ferme pas correctement, notamment lors d'une défaillance du système ou d'une panne secteur. Transformer conserve un maximum d'informations dans un objet SuspendedModel de la collection SuspendedModels.

Si le modèle est corrompu, utilisez la méthode Remove ou Delete pour le supprimer de la collection. S'il est récupérable, utilisez la méthode OpenModel et affectez au paramètre NomFichier la valeur de retour de la propriété QyPath.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Booléen

### Accès

Lecture

### Exemples

```
objSuspendedModel = objTransApp.SuspendedModels(intX)If objSuspendedModel.IsBad  
= True Then
```

---

## Propriété IsExpressionValid

La propriété IsExpressionValid renvoie une valeur indiquant si une expression est valide.

### Syntaxe

*object* .IsExpressionValid

## Application

Objet CalculationDefinition

Objet Category

Objet Column

Objet Measure

Objet SpecialCategory

## Explication

Vous pouvez ajouter des informations à un modèle qui n'est pas directement basé sur une source de données en créant une expression avec la propriété ExpressionText. Transformer évalue toutes les expressions quand l'objet parent est mis à jour ainsi qu'à l'exécution. Pour éviter toute erreur inattendue, utilisez IsExpressionValid pour déterminer si tous les composants de l'expression sont valides.

## Type

Booléen

## Accès

Lecture

---

## Propriété IsFolder

La propriété IsFolder définit ou retourne une valeur indiquant si une mesure représente un dossier de mesures.

## Syntaxe

*Measure* .IsFolder

## Application

Objet Measure

## Explication

Utilisez cette propriété pour changer une mesure en un dossier de mesures, ou l'inverse.

## Type

Booléen

## Accès

Lecture/écriture



---

## Propriété IsManual

La propriété IsManual retourne une valeur indiquant si un niveau est associé à une valeur source.

### Syntaxe

*object* .IsManual

### Application

Objet DateLevel

Objet de niveau

### Explication

Un niveau manuel n'est pas associé à une valeur source. Vous le créez plutôt manuellement comme niveau intermédiaire pour regrouper un grand nombre de catégories enfants ou des catégories basées sur un attribut différent de ceux qui figurent dans vos sources de données.

Si une source de données contient des colonnes pour l'état et la ville, par exemple, vous pouvez créer un niveau manuel ainsi que les catégories requises pour regrouper les villes par région géographique plutôt que par état ou par province.

Les niveaux manuels ne possèdent pas de colonne directement associée à la source et se connectent généralement à des catégories enfants.

La valeur True indique un niveau manuel. La valeur False indique un niveau source.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Booléen

### Accès

Lecture

---

## IPropriété sMDCInUse

La propriété IsMDCInUse retourne une valeur qui indique si un cube est en cours d'utilisation ou de construction.

### Syntaxe

*object* .IsMDCInUse

## Application

Objet Cube

Objet CubeGroup

Objet ChildCube

## Explication

Un cube est verrouillé s'il est en cours d'utilisation par un client PowerPlay ou s'il est en cours de mise à jour. Utilisez IsMDCInUse pour vérifier si le cube est ouvert.

Par exemple, si un cube est ouvert dans une autre application client de PowerPlay ou si un utilisateur tente de construire le cube, la propriété IsMDCInUse retournera la valeur True.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Booléen

## Accès

Lecture

---

## Propriété IsolationLevel

La propriété IsolationLevel définit ou retourne les types de transaction autorisés quand vous référencez un fichier de définition de requête d'Impromptu (.iqd).

## Syntaxe

*IqdDataSource* .IsolationLevel

## Application

Objet IqdDataSource

## Explication

Affectez à cette propriété un des paramètres de transaction ci-dessous.

Valeur de type long	Description
0	Par défaut. Utilise le niveau d'isolement spécifié initialement lors de la création de la source de données.
1	ReadUncommitted. Fait en sorte qu'une transaction ait immédiatement accès aux changements apportés par d'autres transactions.

Valeur de type long	Description
2	ReadCommitted. Ce mode permet à une transaction d'accéder uniquement aux lignes qui ont été confirmées par d'autres transactions.
3	CursorStability. Ce mode interdit aux autres transactions de mettre à jour la ligne sur laquelle se trouve la transaction.
4	ReproducibleRead. Veille à ce que les lignes sélectionnées ou mises à jour par une transaction ne soient pas modifiées par une autre transaction tant que la première n'est pas terminée.
5	PhantomProtection. Interdit à une transaction l'accès aux lignes insérées ou supprimées depuis le début de la transaction.
6	Serializable. Permet de s'assurer qu'un ensemble de transactions exécutées simultanément produise le même résultat que si elles étaient exécutées séquentiellement.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
datasource
= model.DataSources.Add(xtrObjectType.trIqdDataSource)datasource.IsolationLevel
= 0
```

---

## Propriété IsPrimary

La propriété IsPrimary définit ou retourne une valeur indiquant si un objet est ou non la catégorie de branche principale.

## Syntaxe

*object* .IsPrimary

## Application

Objet DateDrillDown

Objet DrillDown

Objet Category

## Explication

Cette propriété prend la valeur True pour la première branche de niveaux d'une collection DrillDowns et la valeur False pour toutes les autres. False désigne une

branche de niveaux secondaire. Des branches de niveaux secondaires sont créées à l'aide de la méthode `CreateAlternateDrillDown`.

Par exemple, deux branches sont disponibles dans la dimension Réseaux: la branche principale est la branche Type de réseau et la branche secondaire, la branche Par région. Si votre branche de niveaux est la branche principale, vous passez de Type de réseau à Client. S'il s'agit de la branche secondaire, vous passez de Région à Client.

La propriété `IsPrimary` ne peut pas être modifiée pour un objet `Category`. S'il s'agit d'un objet `DateDrillDown` ou `DrillDown`, la valeur `False` peut devenir `True`, mais pas l'inverse. Quand vous affectez la valeur `True` à une branche de niveaux, elle bascule d'une branche secondaire vers une branche principale et la branche de niveaux principale précédente devient une branche secondaire.

Dans `Transformer`, les allocations et les découpages en partitions sont basés sur la branche principale de niveaux. Dans `PowerPlay`, les valeurs de cumul sur catégorie récapitulent jusqu'au niveau supérieur de la branche principale de niveaux.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Booléen

## Accès

Lecture (`Category`)

Lecture/Ecriture (`DateDrillDown` et `DrillDown`)

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")
objLevel = objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjAltDrill.IsPrimary = True
```

---

## Propriété `IsTimeBasedPartitionedCube`

La propriété `IsTimeBasedPartitionCube` définit ou retourne une valeur qui indique si un groupe de cubes est défini comme cube découpé en partitions de temps.

## Syntaxe

`CubeGroup` .`IsTimeBasedPartitionedCube`

## Application

Objet `CubeGroup`

## Explication

Pour créer un cube découpé en partitions de temps, définir un objet `CubeGroup` puis définir la valeur de la propriété `IsTimeBasedPartitionedCube` à `True`. Si la valeur a été définie à `True`, le fait de la définir à `False` occasionnera une exception.

Il est possible de définir le niveau et la dimension de la même manière que l'objet CubeGroup. Cependant, on y accède par l'intermédiaire du cube découpé en partitions de temps.

**Paramètre par défaut** : False

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Booléen

### **Accès**

Lectures multiples/Ecriture unique

---

## **Propriété KeyName**

La propriété KeyName définit ou retourne la valeur qui apparaît dans la colonne de la source de données associée.

### **Syntaxe**

*Category* .KeyName

### **Application**

Objet Category

### **Explication**

Cette valeur doit être unique parmi toutes les catégories appartenant au parent immédiat de la catégorie. Pour les niveaux de convergence, cette valeur doit être unique parmi toutes les catégories du niveau.

### **Type**

Chaîne

### **Accès**

Lecture/écriture

### **Exemples**

```
objCategory = objModel.Dimensions("Sales region").Drilldowns(1).Categories(2)If  
objCategory.KeyName <> "Central Europe" Then
```

---

## **Propriété Label**

La propriété Label définit ou retourne un nom descriptif qui apparaît dans PowerPlay.

### **Syntaxe**

*object* .Label

## Application

Objet Association

Objet Category

Objet CategorySet

Objet CurrencyRecord

Objet Measure

Objet SpecialCategory

## Explication

Le libellé par défaut d'une catégorie ordinaire est la propriété Name.

Le libellé par défaut d'un enregistrement de devise est le code de pays ou de région, tel qu'il est défini par la propriété CountryCode.

Le libellé par défaut d'une catégorie spéciale est la propriété Name. Le libellé par défaut des autres objets est la propriété Name. Le libellé d'une association est le nom de l'objet de référence.

Les libellés ne doivent pas nécessairement être uniques.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objAssociation = objDrill.ConvergenceLevel.Associations.Add()objAssociation.Label  
= objColumn.Name
```

---

## Propriété LastUseDate

La propriété LastUseDate retourne la date à laquelle la catégorie ou la catégorie spéciale a été activée pour la dernière fois.

## Syntaxe

*object* .LastUseDate

## Application

Objet Category

Objet SpecialCategory

## Explication

Utilisez cette propriété pour afficher la date à laquelle Transformer a généré une catégorie ou a modifié des propriétés de la catégorie.

Utilisez la valeur de retour de cette propriété avec la méthode CleanHouse pour supprimer les catégories inactives. Si la date de la propriété LastUseDate est antérieure à celle du paramètre Date de la propriété CleanHouse, Transformer considère la catégorie comme inactive et la supprime.

## Type

Long

## Accès

Lecture

---

## Propriété LatestDate

La propriété LatestDate définit la date la plus ancienne dans une plage destinée à la sélection de catégories.

## Syntaxe

*object* .LatestDate

## Application

Objet DateDimension

Objet DateWizard

## Explication

Utilisez cette propriété avec la propriété EarliestDate pour limiter la plage des catégories de date acceptables dans une dimension de temps. La définition de limites permet de supprimer des catégories qui ne sont pas significatives pour les utilisateurs de PowerPlay, telles que les années antérieures.

Transformer met automatiquement à jour toutes les catégories quand les catégories du cube sont générées ou qu'un cube est créé. Pour générer des catégories de date uniquement, utilisez la propriété GenerateTimePeriod (objet DateDimension). Transformer ne génère qu'un sous-ensemble des enregistrements compris entre les valeurs des propriétés EarliestDate et LatestDate.

Si Transformer rencontre des valeurs de date qui ne figurent pas dans la plage spécifiée, il génère une catégorie Early Dates et/ou Late Dates, selon l'échéance des dates qui ne sont pas comprises dans la plage. Si Transformer ne rencontre aucune date antérieure ou postérieure et si les dates ne peuvent pas être placées dans la plage spécifiée, il génère une catégorie Invalid Dates.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/Ecriture (DateDimension)

Ecriture (DateWizard)

## Exemples

.LatestDate = 19941231

---

## Propriété Level

La propriété Level renvoie un niveau pour un objet Category ou SpecialCategory.

## Syntaxe

*Category* .Level

*SpecialCategory* .Level

## Application

Objet Category

Objet SpecialCategory

## Explication

Utilisez cette propriété pour trouver à quels niveaux une catégorie ou une catégorie spéciale appartient. Les catégories ou catégories spéciales n'appartiennent qu'à un seul niveau.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - Level

## Accès

Lecture

---

## Propriété LevelCategories

La propriété LevelCategories renvoie une collection de catégories dans un niveau.

## Syntaxe

*Object* .LevelCategories



## Application

Objet DateLevel

Objet de niveau

## Explication

Utilisez cette propriété pour accéder à la collection de catégories présentes dans un niveau. Une catégorie ne peut appartenir qu'à un seul niveau.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - LevelCategories

## Accès

Lecture

---

## Propriété LevelDrillDowns

La propriété LevelDrillDowns retourne la collection d'objets de la branche de niveaux à laquelle appartient le niveau.

## Syntaxe

*object* .LevelDrillDowns

## Application

Objet DateLevel

Objet de niveau

## Explication

Comme la plupart des niveaux possèdent une seule branche de niveaux, la collection LevelDrillDowns ne renferme généralement qu'un seul objet DrillDown ou DateDrillDown. Quand un niveau de branches correspond au niveau de convergence de deux branches de niveaux ou davantage, la collection possède un nombre d'entrées équivalent.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - LevelDrillDowns

## Accès

Lecture

## Exemples

```
objDateDrillDown = objModel.Dimensions("Years").DrillDowns(1)
```

---

## Propriété Levels

La propriété Levels renvoie une collection d'objets de niveau ou DateLevel.

### Syntaxe

*object* .Levels

### Application

Objet DateDrillDown

Objet DrillDown

### Explication

Chaque dimension d'un modèle contient une ou plusieurs branches de niveaux, possédant chacune une collection Levels.

La méthode CreateDateDimension crée automatiquement une collection de niveaux de date. La méthode DoAutoDesign crée automatiquement une collection de niveaux pour chaque branche de niveaux d'un modèle.

Une collection Levels contient des objets de niveau ou des objets DateLevel, mais pas les deux.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet - Levels

### Accès

Lecture

### Exemples

```
objTimeDimension = objModel.Dimensions.Item("Time")objLevel  
= objTimeDimension.DrillDowns.Item(1).Levels.Item("Month")
```

---

## Propriété LocalPath

La propriété LocalPath définit ou retourne l'emplacement de la source de données.

### Syntaxe

*object* .LocalPath

## Application

Objet CrossTabDataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

Cette propriété peut inclure un chemin relatif et le nom de fichier ou un emplacement complet. Si la source de données figure sur un serveur, utilisez plutôt la propriété ServerPath.

Si vous définissez un chemin d'accès relatif, Transformer l'ajoute à la valeur de départ du chemin spécifiée dans la propriété DataSource. Si vous ne spécifiez pas la valeur de la propriété DataSource, Transformer ajoute la valeur de la propriété LocalPath à l'emplacement dans lequel est installé PowerPlay. Dans l'un ou l'autre cas, les valeurs concaténées doivent donner comme résultat un emplacement correctement défini.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources.
  Add(xtrObjectType.trFlatFileDataSource)objDataSource.LocalPath
= strDataPath
```

---

## Propriété LogErrorLevel

La propriété LogErrorLevel définit ou retourne le niveau de gravité des messages d'erreur consignés.

## Syntaxe

*Application* .LogErrorLevel

## Application

Objet Application

## Explication

Utilisez cette propriété pour spécifier le niveau de message ou d'avertissement qui est enregistré. La constante trLogInformationAndAbove, par exemple, consigne tous les messages dans le fichier. Pour consigner des messages, vous devez d'abord affecter la valeur True à la propriété EnableMessageLogging.

Cette propriété utilise les valeurs de la constante `xtrPreferences`.

### Type

Constante

### Accès

Lecture/écriture

### Exemples

```
objTransApp.LogErrorLevel = xtrPreferences.trLogErrorsAndAbove
```

---

## Propriété `LogFileAppend`

La propriété `LogFileAppend` définit ou retourne une valeur indiquant si Transformer ajoute des messages au fichier journal ou s'il écrase les messages de journal.

### Syntaxe

*Application* .`LogFileAppend`

### Application

Objet `Application`

### Explication

Quand vous affectez la valeur `True` à cette propriété, Transformer ajoute des informations au fichier journal. Si vous affectez la valeur `False`, Transformer écrase le fichier journal.

Utilisez les propriétés `LogFileName` et `LogFilePath` pour spécifier le nom et l'emplacement d'un fichier journal. En l'absence de spécification, Transformer crée un fichier dans le dossier où sont installées les applications PowerPlay.

Pour consigner des messages, vous devez affecter la valeur `True` à la propriété `EnableMessageLogging`.

**Paramètre par défaut :** `False`

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objTransApp.LogFileAppend = True
```

---

## Propriété LogFileName

La propriété LogFileName définit ou retourne le nom du fichier journal.

### Syntaxe

*Application* .LogFileName

### Application

Objet Application

### Explication

Si rien n'est indiqué, Transformer utilise le même nom que celui du modèle en cours mais lui donne une extension de fichier .log. Si aucun nom de modèle en cours n'est fourni, Transformer génère un nom arbitraire, par exemple Tfmr001e.log.

Utilisez la propriété LogFilesPath pour définir l'emplacement dans lequel les fichiers journaux sont consignés. Pour consigner des messages, vous devez d'abord affecter la valeur True à la propriété EnableMessageLogging.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objTransApp.LogFileName = "TrModelsLog.log"
```

---

## Propriété LogFilesPath

La propriété LogFilesPath définit ou retourne l'emplacement dans lequel Transformer enregistre le fichier journal.

### Syntaxe

*Application* .LogFilesPath

### Application

Objet Application

### Explication

Utilisez cette propriété pour indiquer à Transformer à quel emplacement les fichiers journaux doivent être sauvegardés. Si vous ne définissez aucun emplacement, Transformer utilise celui spécifié par la propriété ModelsPath. Si la propriété ModelsPath n'est pas définie, Transformer stocke le fichier journal dans le répertoire d'installation de PowerPlay.

Utilisez la propriété `LogFileName` pour spécifier le nom du fichier journal. Pour consigner des messages, vous devez d'abord affecter la valeur `True` à la propriété `EnableMessageLogging`.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objTransApp.LogFilePath = strStartLocation
```

---

## Propriété Lunar

La propriété `Lunar` définit ou retourne une valeur indiquant si l'objet `DateDrillDown` est basé sur une année lunaire.

### Syntaxe

```
DateDrillDown .Lunar
```

### Application

Objet `DateDrillDown`

### Explication

Le type de calendrier que représente un objet `DateDrillDown`, à savoir ordinaire ou lunaire, dépend de la nature de la dimension de temps parent. La valeur `False` désigne une année calendaire.

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objDateWizard  
= objModel.DateWizardobjDateDim  
= objDateWizard.CreateDateDimension()objDrillDown  
= objDateDim.DrillDowns(1)objDrillDown.Lunar  
= True
```

---

## Propriété ManualCurrentPeriod

La propriété `ManualCurrentPeriod` définit ou retourne une valeur indiquant si la période de date en cours est définie manuellement ou par `Transformer`.

## Syntaxe

*DateDimension* .ManualCurrentPeriod

## Application

Objet DateDimension

## Explication

Quand cette propriété a la valeur True, la période en cours peut correspondre à n'importe quelle catégorie de la dimension de temps, généralement celle du niveau le plus bas de la dimension, notamment sous un niveau hebdomadaire.

Les catégories spéciales utilisent la période de date en cours pour créer des périodes de date relatives telles que le mois Current, le mois dernier et le trimestre à ce jour.

Quand la valeur est False, Transformer définit la période de date en cours à l'aide de la catégorie qui possède la date la plus récente. La propriété SetsCurrentPeriod désigne la source de données qui fournit la période en cours. Si la seule catégorie détectée est Dates anciennes, Transformer ne définit aucune période en cours pour la dimension.

**Valeur par défaut :** False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDateDim = objModel.Dimensions("Date")MsgBox("Current  
Period:" + objDateDim.ManualCurrentPeriod.Name)
```

---

## Propriété MaximizeSpeed

La propriété MaximizeSpeed définit ou renvoie une valeur indiquant si Transformer vérifie le caractère unique d'une catégorie.

## Syntaxe

*object* .MaximizeSpeed

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

Utilisez cette propriété pour optimiser la vitesse. Quand la valeur est True, Transformer ne vérifie pas le caractère unique des catégories lorsqu'il remplit un niveau dans une dimension.

Quand la valeur est False, Transformer tente de détecter les problèmes d'unicité dans les définitions de niveaux. Dans Transformer, des niveaux sont uniques quand la valeur de chaque catégorie d'un niveau est différente des autres.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDataSource  
= objModel.DataSources.Add(xtrObjectType  
.trFlatFileDataSource)objDataSource.MaximizeSpeed  
= True
```

---

## Propriété MaxNumPartLevels

La propriété MaxNumPartLevels définit ou retourne le nombre maximum de niveaux de partition.

## Syntaxe

*object* .MaxNumPartLevels

## Application

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété pour définir le nombre maximum de fois que Transformer lit les données sources. Etant donné que le nombre de niveaux de partitions augmente, Transformer doit consulter plus fréquemment les données.

Si la propriété Optimize a la valeur Autopartition, vous devez également définir les propriétés du cube : DesiredPartitionSize et EstimatedRows.

Pour créer manuellement des partitions, utilisez la propriété Partition d'un niveau ou d'une catégorie. Assurez-vous que la valeur de la propriété MaxNumPartLevels est au moins égale au nombre de niveaux de partition ajoutés.



**Paramètre par défaut :** 5 transmissions

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Long

### **Accès**

Lecture/écriture

### **Exemples**

```
objCube = objModel.Cubes.Item(1)objCube.MaxNumPartLevels  
= 5
```

---

## **Propriété MaxTransactionNumber**

La propriété MaxTransactionNumber définit ou retourne le nombre maximum d'enregistrements que Transformer traite avant d'enregistrer les modifications dans un cube.

### **Syntaxe**

*Application* .MaxTransactionNumber

### **Application**

Objet Application

### **Explication**

Utilisez cette propriété pour déterminer à quel moment Transformer insère les points de contrôle aux différentes étapes qui ponctuent le processus de génération de cubes. Ces paramètres limitent le nombre d'enregistrements gardés en statut temporaire avant l'insertion d'un point de contrôle. Si une erreur venait à empêcher la création d'un cube, vous pouvez redémarrer le processus au dernier point de contrôle enregistré.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

**Paramètre par défaut :** 500 000

### **Type**

Long

### **Accès**

Lecture/écriture

### **Exemples**

```
objTransApp.MaxTransactionNumber = 500000
```

---

## Propriété MDCFile

La propriété MDCFile définit ou retourne le nom d'un fichier de PowerCube (.mdc).

### Syntaxe

*object* .MDCFile

### Application

Objet ChildCube

Objet Cube

Objet CubeGroup

### Explication

Cette propriété permet de spécifier un nom de fichier.

Pour spécifier le dossier dans lequel le fichier de PowerCube doit être enregistré, utilisez la propriété PowerCubesPath. Transformer recherche la propriété PowerCubesPath, puis la propriété ModelsPath pour trouver un emplacement. Si aucun emplacement n'est spécifié, Transformer enregistre le fichier du PowerCube à l'emplacement où vous avez installé PowerPlay.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objCube = objModel.Cubes.Item(1)objCube.MDCFile = "GoCube"
```

---

## Propriété MeasureInclude

La propriété MeasureInclude définit ou retourne une valeur indiquant si une mesure est incluse dans l'objet spécifié.

### Syntaxe

*object* .MeasureInclude(Measure)

### Application

Objet Cube

Objet CubeGroup

Objet CustomView

## Explication

Pour inclure une mesure spécifique, utilisez le nom comme paramètre et affectez la valeur True à la propriété. Pour exclure une mesure, affectez la valeur False à cette propriété. Pour définir le statut d'inclusion de plusieurs mesures, utilisez cette propriété à plusieurs reprises en nommant chaque fois une mesure différente dans le paramètre.

Quand vous excluez une mesure utilisée dans l'expression d'une mesure calculée qui est elle-même incluse dans le cube, Transformer inclut la mesure dans le cube pour le calcul, mais ne la rend pas visible aux yeux des utilisateurs de PowerPlay.

Paramètre par défaut : True

Paramètre	Description
Mesure	Requise. Spécifie l'objet Measure qui est inclus ou exclu de l'objet désigné.  Type : Objet

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion  
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion  
  .MeasureInclude(objModel.Measures("Product cost"))  
= False
```

---

## Propriété MeasureName

La propriété MeasureName définit ou retourne un titre descriptif qui identifie une mesure.

## Syntaxe

*object* .MeasureName

## Application

Objet ChildCube

Objet Cube

Objet CubeGroup

## Explication

Utilisez cette propriété pour remplacer le titre par défaut "Mesure" qui apparaît sur la ligne des dimensions de PowerPlay.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion  
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion.MeasureName  
= "Revenue Made"
```

---

## Propriété Measures

La propriété Measures renvoie une collection d'objets Measure.

## Syntaxe

*Model* .Measures

## Application

Objet Model

## Explication

Une collection Measures peut contenir des mesures ordinaires, des mesures calculées et des mesures de nombre de catégories.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - Measures

## Accès

Lecture

## Exemples

```
objMeasures = objModel.Measures
```

---

## Propriété MeasureType

La propriété MeasureType renvoie le type d'un objet Measure.

## Syntaxe

*Measure* .MeasureType

## Application

Objet Measure

## Explication

Une collection Measures peut contenir des mesures ordinaires, des mesures calculées et des mesures de nombre de catégories. Le type `trRegularMeasure` est automatiquement sélectionné quand vous utilisez la méthode `DoAutoDesign`. Une mesure ordinaire est associée à une colonne ou à un attribut.

Vous pouvez utiliser la propriété `ExpressionText` pour définir une mesure calculée et la propriété `CategoryCountLevel` pour définir une mesure de nombre de catégories.

La propriété `MeasureType` utilise les valeurs de la constante `xtrMeasureType`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - `xtrMeasureType`

## Accès

Lecture

---

## Propriété MissingValue

La propriété `MissingValue` définit ou retourne la valeur de remplacement à utiliser quand la valeur numérique d'une mesure manque dans la source de données.

## Syntaxe

*Measure* .`MissingValue`

## Application

Objet Measure

## Explication

Par défaut, toutes les valeurs manquantes sont affichées sous forme de zéros dans `PowerPlay`. `PowerPlay` peut afficher le message "nd" à la place des zéros. De cette façon, les éléments manquants ne sont pas considérés comme des valeurs nulles.

La propriété `MissingValue` utilise les valeurs de la constante `xtrMissingValue`.

## Type

Constante - `xtrMissingValue`

## Accès

Lecture/écriture

## Exemples

```
objMeasures  
= objModel.MeasurescurrentMeasure  
= objMeasures("Revenue")currentMeasure.MissingValue  
= xtrMissingValue.trMissingValueZERO
```

---

## Propriété ModelName

La propriété ModelName définit ou renvoie le nom d'un fichier modèle pour lequel un fichier temporaire (.qy?) a été créé à la suite d'une fin anormale.

### Syntaxe

*SuspendedModel* .**ModelName**

### Application

Objet SuspendedModel

### Explication

Lorsqu'un modèle client n'a pas été enregistré en raison d'un arrêt anormal, vous pouvez utiliser la propriété SuspendedModels pour renvoyer la collection des modèles en attente. La méthode Item permet de renvoyer chaque modèle, et la propriété IsBad, de vérifier si un modèle est endommagé.

Si vous n'avez pas nommé le modèle avant la défaillance, cette propriété renvoie le nom "Nouveau modèle".

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objSuspendedModel = objTransApp.SuspendedModels(intX)MsgBox("Model:  
" + objSuspendedModel.ModelName + "_", located at " + objSuspendedModel.QyPath  
+ " is corrupt")
```

---

## Propriété ModelsPath

La propriété ModelsPath définit ou retourne l'emplacement dans lequel Transformer ouvre et enregistre les fichiers modèles.

### Syntaxe

*Application* .**ModelsPath**

### Application

Objet Application

## Explication

Si aucun emplacement n'est spécifié, les fichiers modèles sont stockés dans le répertoire d'installation de PowerPlay.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objTransApp.ModelsPath = strStartLocation
```

---

## Propriété ModelTemporaryFilePath

La propriété ModelTemporaryFilePath définit ou retourne l'emplacement dans lequel Transformer crée des fichiers modèles temporaires (.qy?).

## Syntaxe

*Application* .ModelTemporaryFilePath

## Application

Objet Application

## Explication

Utilisez des fichiers modèles temporaires pour récupérer un modèle en attente en présence d'une erreur fatale. Si cette propriété n'est pas définie, Transformer recherche l'emplacement (dans l'ordre suivant) :

- spécifié par la propriété ModelsPath,
- où PowerPlay est installé
- défini par la variable Temporary = dans la section [Services] du fichier cs7g.ini
- défini par la variable d'environnement TEMP,
- défini par la variable d'environnement TMP.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objTransApp.ModelTemporaryFilePath = strStartLocation
```

---

## Propriété ModelType

La propriété ModelType renvoie l'extension d'un fichier modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.

### Syntaxe

*Model* .ModelType

### Application

Objet Model

### Explication

Utilisez cette propriété pour vérifier le type d'un fichier modèle ou pour consigner le type dans un fichier journal. Le modèle peut être enregistré dans un fichier texte (.mdl) ou un fichier binaire (.py?). Transformer remplace le point d'interrogation par un nombre ou une lettre.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Chaîne

### Accès

Lecture

---

## Propriété MonthType

La propriété MonthType définit le mode de calcul du niveau mois d'une dimension de temps.

### Syntaxe

*DateWizard* .MonthType

### Application

Objet DateWizard

### Explication

Cette propriété permet de déterminer si un mois fait partie d'une année calendaire ou d'une année lunaire.

La propriété MonthType utilise les valeurs de la constante xtrSpecialFunction.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



## Type

Constante - `xtrSpecialFunction`

## Accès

Ecriture

## Exemples

```
objModel.MonthType = xtrSpecialFunction.trSpecialFunctionMonth
```

---

## Propriété Name

La propriété Name définit ou renvoie le nom d'un objet.

## Syntaxe

*objet* .Name

## Application

«Objet Application», à la page 45

Objet Association

Objet CalculationDefinition

Objet Category

Objet CategorySet

Objet ChildCube

Objet Column

Objet CrossTabDataSource

Objet Cube

Objet CubeGroup

Objet CurrencyTable

Objet CustomView

Objet DataSource

Objet DateDimension

Objet DateDrillDown

Objet DateLevel

Objet DbDataSource

Objet Dimension

Objet DrillDown  
Objet DrillThroughTarget  
Objet Filter  
Objet FlatFileDataSource  
Objet IqdDataSource  
Objet de niveau  
Objet Measure  
Objet Model  
Objet Name  
Objet Namespace  
Objet Package  
Objet Prompt  
Objet Query  
«Objet Report», à la page 113  
Objet SecurityObject  
Objet de code d'accès  
Objet SpecialCategory  
Objet View

## **Explication**

Pour créer un objet dimension, vous devez spécifier un nom.

Certains objets possèdent des propriétés Name et Label ou Name et ShortName. Si vous ne spécifiez pas une valeur Label ou ShortName pour un objet, Transformer utilise la propriété Name comme source par défaut.

Si le nom d'un nouvel objet ne possède pas de source pour les valeurs de nom, Transformer en crée une et lui ajoute un identificateur unique. Par exemple, Vue Produits1, Vue Produits2.

Dans le cas des objets Category, Transformer utilise la propriété KeyName comme source par défaut pour les propriétés Name et Code.

## **Type**

Chaîne

## Accès

Lecture (Application, Name et Report)

Lecture/Ecriture (tous les autres objets)

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.Name = "Great  
Outdoors Sales (Optimized)"
```

---

## Propriété Namespaces

La propriété Namespaces renvoie une collection Namespaces.

### Syntaxe

*Model* .Namespaces

### Application

Objet Model

### Explication

Cette propriété permet d'effectuer une itération dans les espaces-noms d'un objet Model, d'ajouter un nouvel espace-noms et de modifier ou supprimer un espace-noms.

### Type

Objet - Namespaces

### Accès

Lecture

### Exemples

```
new_namespace = model.Namespaces.Add()
```

---

## Propriété NewCatsLocked

La propriété NewCatsLocked définit ou retourne une valeur indiquant si vous pouvez ajouter de nouvelles catégories à l'objet.

### Syntaxe

*object* .NewCatsLocked

### Application

Objet DateDimension

Objet DateLevel

Objet Dimension

Objet de niveau

## Explication

Cette propriété permet de verrouiller un niveau ou une dimension de manière à ce qu'aucune nouvelle catégorie ne soit ajoutée lors d'une génération de catégorie. Quand cette propriété a la valeur True pour une dimension, elle prend aussi la valeur True pour tous les niveaux de cette dimension.

Si la valeur est True, Transformer ignore les valeurs sources de la source de données qui ne sont pas liées aux catégories déjà présentes dans le modèle.

Valeur par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")objLevel  
= objLocationsDim.DrillDowns(1).Levels("Employee")objLevel.NewCatsLocked  
= False
```

---

## Propriété ObjectCAMID

La propriété ObjectCAMID renvoie le CAMID de l'objet dans l'espace-noms défini par la propriété ObjectName.

## Syntaxe

*Namespace* .ObjectCAMID

## Application

Objet Namespace

## Explication

Cette propriété permet d'extraire le CAMID d'un objet dans un espace-noms.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

## Exemples

```
new_namespace = model.Namespaces.Add()CAMID_of_Object  
= new_namespace.ObjectCAMID
```

---

## Propriété ObjectName

La propriété ObjectName définit ou renvoie le nom d'un objet Namespace.

### Syntaxe

*Namespace* .ObjectName

### Application

Objet Namespace

### Explication

Cette propriété permet de définir le nom d'un objet Namespace pour extraire le CAMID de l'objet de la propriété ObjectCAMID.

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
new_namespace = model.Namespaces.Add()new_namespace.ObjectName  
= Name_of_Object 'Authors is a group
```

---

## Propriété Optimize

La propriété Optimize définit ou retourne l'option d'optimisation du cube en cours pour votre modèle et l'environnement.

### Syntaxe

*object* .Optimize

### Application

Objet ChildCube

Objet Cube

Objet CubeGroup

## Explication

Les cubes sont optimisés pour améliorer les performances dans Transformer et PowerPlay. Vous pouvez, par exemple, spécifier la constante `trOptimizeDataPasses` pour optimiser le nombre de passes dans les fichiers de travail temporaires pendant la création d'un cube.

Si cette propriété a pour valeur `trOptimizeAutoPartition`, définissez les propriétés du cube: `DesiredPartitionSize`, `EstimatedRows` et `MaxNumPartLevels`. Vous pouvez utiliser la propriété `ExcludeAutoPartition` pour exclure une dimension du processus de découpage automatique en partitions.

La propriété `Optimize` utilise les valeurs de la constante `xtrCubeOptimize`.

## Type

Constante - `xtrCubeOptimize`

## Accès

Lecture/écriture

## Exemples

```
objCube = objModel.Cubes.Item(1)objCube.Optimize  
= xtrCubeOptimize.trOptimizeAutoPartition
```

---

## Propriété `OrderByDescending`

La propriété `OrderByDescending` définit ou retourne une valeur indiquant si les catégories sont triées dans l'ordre croissant.

## Syntaxe

```
object .OrderByDescending(DrillDown)
```

## Application

Objet `DateLevel`

Objet de niveau

## Explication

Quand un niveau représente la convergence de deux branches de niveaux ou davantage, vous pouvez appliquer un ordre de tri différent à chacune d'elles. Utilisez le paramètre `DrillDown` pour spécifier la branche de niveaux à laquelle s'applique le tri.

Par défaut, les catégories sont triées dans l'ordre croissant.

**Paramètre par défaut :** `False`

Paramètre	Description
DrillDown	Requise. Spécifie la structure des branches de niveaux dans lesquelles le niveau apparaît.  Type : Variante

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDrill
= objModel.Dimensions("Retailers").DrillDowns(2)objDrill
.Levels("Retailer site").OrderByDescending(objAssociation.Context)
= False
```

---

## Propriété OrderByStorageType

La propriété OrderByStorage définit ou retourne une valeur indiquant la manière dont les catégories sont triées en fonction du type de stockage d'une colonne.

## Syntaxe

*object* .**OrderByStorageType**(DrillDown)

## Application

Objet DateLevel

Objet de niveau

## Explication

Cette propriété permet de définir un classement de catégories quand le tri est basé sur une colonne possédant une classe de données de type numérique. Par exemple, vous basez un tri de niveaux de produits sur la quantité vendue et la mesure appelée "Quantité", qui est associée à la colonne Quantité, possède un stockage de type "entier 16 bits".

La propriété OrderByStorageType utilise les valeurs de la constante xtrStorage.

Paramètre	Description
DrillDown	Requise. Spécifie la structure (contexte) des branches de niveaux dans lesquelles le niveau apparaît.  Type : Variante

## Type

Constante - xtrStorage

## Accès

Lecture/écriture

---

## Propriété Origin

La propriété Origin renvoie l'origine de l'objet spécifié.

### Syntaxe

*Column* .Origin

### Application

Objet Column

### Explication

Un objet attribut ou colonne peut être créé de plusieurs manières. La plupart des objets de colonne, par exemple, sont générés à partir de la source de données par Transformer tandis qu'une colonne calculée est créée manuellement.

La propriété Origin utilise les valeurs de la constante xtrOrigin.

## Type

Constante - xtrOrigin

## Accès

Lecture

---

## Propriété OriginalName

La propriété OriginalName définit ou retourne le nom de la colonne dans la source de données.

### Syntaxe

*Column* .OriginalName

### Application

Objet Column

### Explication

Utilisez cette propriété pour établir des correspondances croisées entre les données de votre modèle et celles de la source de données d'origine.

La valeur de retour dépend du format de la source de données.



Données sources	OriginalName
Fichier de définition de requête d'Impromptu (.iqd)	Nom défini dans la définition de requête
Fichier de texte à zones délimitées avec titres de colonne	Titre de la colonne
Fichier de texte à zones délimitées	Nom composé du préfixe "Things like" et suivi de la valeur dans la zone correspondante du premier enregistrement de la source de données
Fichier de texte à zones fixes	Aucun nom d'origine
Table dBase, Paradox, Lotus 1-2-3, Excel, Clipper ou FoxPro	Nom de colonne défini dans la table
Pack IBM Cognos	Nom défini dans la définition de requête.
Rapport IBM Cognos	Nom défini dans la définition de requête.
Tableau croisé Lotus 1-2-3 ou Excel	Le nom de plage qui représente une colonne dans la source de données d'un tableau.
Sous-fichier transférable de PowerHouse	Nom défini dans le sous-fichier

## Type

Chaîne

## Accès

Lecture

## Exemples

```
new_package
= model.Packages.Add()new_query
= new_package.Queries.Add()new_column1
= new_query.Columns.Add()new_column1.OriginalName
= "[Sales (query)].[Time dimension].[Date]"
```

---

## Propriété Orphanage

La propriété Orphanage définit ou retourne une valeur indiquant si une catégorie est isolée.

## Syntaxe

*Category* .Orphanage

## Application

Objet Category

## Explication

Utilisez un orphelinat avec une dimension connue pour générer des catégories isolées.

Un orphelinat est une catégorie créée dans un niveau manuel. Une fois que l'orphelinat est créé, toute autre catégorie générée qui ne possède pas de position définie dans le modèle devient l'enfant de la catégorie isolée. L'utilisation d'un orphelinat vous permet de conserver de nouvelles données à part jusqu'à ce que vous puissiez les connecter à la catégorie appropriée.

Pour créer un orphelinat, ajoutez un niveau à la collection Levels associée à cette dimension. Si vous souhaitez que l'orphelinat collecte les nouvelles valeurs de catégorie de l'ensemble de la dimension, utilisez la méthode Move pour la déplacer à la première position dans la collection.

Utilisez la méthode Add (Categories) pour ajouter une catégorie à la collection Categories. Assurez-vous que les paramètres de la méthode Add spécifient le type d'objet (trCategory) et le niveau manuel auquel appartient la catégorie. Enfin, affectez la valeur True à la propriété Orphanage.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
.OutputScale = 0
```

---

## Propriété OutputScale

La propriété OutputScale définit ou retourne la valeur d'échelle utilisée pour convertir en valeurs décimales les valeurs d'entier qui apparaissent dans PowerPlay.

## Syntaxe

```
object .OutputScale
```

## Application

Objet Column

Objet Measure

## Explication

Cette propriété permet de définir la puissance de dix par laquelle les valeurs sources sont divisées. Par exemple, avec un paramètre de 2, PowerPlay affiche 95 pour la valeur 9450, c'est-à-dire,  $9450/10^2$  arrondi à 0 décimale.

Pour inclure des décimales dans la valeur de sortie, utilisez la propriété `FormatDecimals`. Dans l'exemple ci-dessus, si la propriété `FormatDecimals` a la valeur 2, `PowerPlay` affiche la valeur 94,50.

Utilisez la propriété `InputScale` pour spécifier la manière dont les valeurs sources sont multipliées lors de leur lecture à partir d'une source de données.

La plage des valeurs d'échelle est comprise entre 0 et 16, ce qui correspond à  $10^{**}16$ .

**Valeur par défaut :** 0

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur. Par exemple, une exception se produit avec `Update()` si la valeur est définie en dehors de la plage allant de 0 à 16.

### Type

Long

### Accès

Lecture/écriture

### Exemples

```
objMeasure.CategoryCountLevel = objLevelobjMeasure.OutputScale  
= 0
```

---

## Propriété Packages

La propriété `Packages` renvoie la collection des objets `Package` associés à un modèle `Transformer`.

### Syntaxe

*Model* .`Packages`

### Application

Objet `Model`

### Explication

Cette propriété permet d'effectuer une itération dans les packs d'un objet `Model`, ainsi que d'ajouter, de modifier ou de supprimer un pack.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Packs

### Accès

Lecture

## Exemples

```
new_package = model.Packages.Add()
```

---

## Propriété PackageDataSourceConnections

La propriété PackageDataSourceConnections renvoie la collection d'objets PackageDataSourceConnection associée à un objet Package ou Report.

### Syntaxe

*object* .PackageDataSourceConnections

### Application

Objet Package

### Explication

Pour utiliser cette propriété, vérifiez qu'il existe au moins une requête affectée au pack ou au rapport.

Cette propriété permet d'effectuer une itération dans les connexions de sources de données pour un pack ou un rapport, d'ajouter de nouvelles connexions, de modifier les propriétés des connexions ou de supprimer des connexions.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

PackageDataSourceConnections

### Accès

Lecture

### Exemples

```
package = model.Packages.Add() connection  
= package.PackageDataSourceConnections.Add()
```

---

## Propriété Parent

La propriété Parent renvoie l'objet Parent.

### Syntaxe

*object* .Parent

### Application

Objet Association

Collection Associations

Objet CalculationDefinition

Collection CalculationDefinitions  
Collection Categories  
Objet Category  
Objet CategorySet  
Collection CategorySets  
Objet ChildCube  
Collection ChildCubes  
Objet Column  
Collection Columns  
Objet CrossTabDataSource  
Objet Cube  
Collection CubeCustomViews  
Objet CubeGroup  
Collection Cubes  
Objet CurrencyRate  
Collection CurrencyRates  
Objet CurrencyRecord  
Collection CurrencyRecords  
Objet CurrencyTable  
Collection CurrencyTables  
Objet CustomView  
collection CustomViews  
Objet DataSource  
Collection DataSources  
Objet DateDimension  
Objet DateDrillDown  
Objet DateLevel  
Objet DateWizard

Objet DbDataSource  
Objet Dimension  
Collection DimensionLevels  
Collection Dimensions  
Objet DrillDown  
Collection DrillDowns  
Objet DrillThroughTarget  
Collection DrillThroughTargets  
Objet Filter  
Collection Filters  
Objet FlatFileDataSource  
Objet IqdDataSource  
Objet de niveau  
collection LevelCategories  
Collection LevelDrillDowns  
Collection Levels  
Objet Measure  
Collection Measures  
Objet Model  
Objet Name  
Collection Names  
Objet Namespace  
Collection Namespaces  
Objet Package  
Collection PackageDatasourceConnections  
Objet PackageDatasourceConnection  
Collection Packages  
Objet Prompt

Collection Prompts  
Collection Queries  
Objet Query  
Objet Report  
Collection Reports  
Objet SecurityObject  
Collection SecurityObjects  
Objet de code d'accès  
Collection Signons  
Objet SpecialCategory  
Objet SuspendedModel  
Collection SuspendedModels  
Objet View  
Collection Views

### **Explication**

Utilisez cette propriété pour retourner l'ancêtre immédiat d'un objet ou d'une collection. Vous pouvez avoir recours à la notation à point pour accéder aux propriétés d'un objet parent et les mettre à jour.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### **Type**

Objet

### **Accès**

Lecture

### **Exemples**

```
objView = objDimension.Views.Add()objView.Parent.Update()
```

---

## **Propriété ParentCategories**

La propriété ParentCategories renvoie une collection de catégories parent.

### **Syntaxe**

*object* .ParentCategories

## Application

Objet Category

Objet SpecialCategory

## Explication

Si une catégorie figure dans plusieurs branches de niveaux, elle peut avoir deux catégories parents ou davantage. Chaque catégorie parent est associée à un niveau différent et à une branche de niveaux distincte.

## Type

Objet

## Accès

Lecture

---

## Propriété Partition

La propriété Partition définit ou renvoie le numéro d'une partition manuelle.

## Syntaxe

*object* .Partition

## Application

Objet Category

Objet DateLevel

Objet de niveau

## Explication

La propriété Partition spécifie le numéro du niveau de partition et définit la (les) catégorie(s) du niveau en tant qu'en-tête d'une partition dans un cube. Les catégories les plus favorables au découpage en partitions sont celles qui possèdent une profondeur de plusieurs niveaux, avec des ratios catégorie/niveau similaires.

Si vous connaissez très bien vos données sources et les conditions requises par les utilisateurs, vous pouvez définir manuellement vos partitions en affectant des numéros de niveau de partition aux niveaux et aux catégories dans des dimensions spécifiques. Si tel est le cas, choisissez des dimensions contenant un grand nombre de catégories et de niveaux ainsi que des ratios catégorie/niveau similaires. Évitez de découper en partitions des dimensions qui contiennent des structures avec des branches secondaires ou des niveaux renfermant des catégories spéciales. Ne spécifiez pas de numéro de partition pour des catégories feuilles, des catégories de branche ou la catégorie racine.

Quand vous créez des partitions, utilisez d'abord la méthode ResetPartitions pour supprimer toutes les partitions automatiques ou manuelles précédentes. Créez ensuite les niveaux de partition à l'aide de la propriété Partition. Utilisez la



propriété MaxNumPartLevels pour spécifier le nombre de passages que Transformer effectue dans les données sources.

Les catégories qui sont supprimées ou filtrées par les valeurs de la constante xtrInclusion ne font pas partie de la partition.

Quand un PowerCube fait l'objet d'une mise à jour incrémentielle, vous ne pouvez pas ajouter de nouveaux niveaux de partition. Afin de modifier le découpage en partitions du modèle, vous devez effectuer un nouveau découpage du modèle et recréer entièrement le PowerCube, en utilisant toutes les données de l'ensemble des incréments.

Plage : 0 à 15.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

### Accès

Lecture/écriture

### Exemples

```
objCategory  
= objModel.Dimensions("Sales region").Drilldowns(1).  
  Categories(2)objCategory.Partition  
= 1
```

---

## Propriété Password

La propriété Password définit un mot de passe prenant en compte la distinction majuscule/minuscule.

### Syntaxe

*object* .Password

### Application

Objet ChildCube

Objet Cube

Objet CubeGroup

Objet de code d'accès

### Explication

Quand un mot de passe est défini au niveau du noeud racine d'un groupe de cubes, le même mot de passe s'applique à tous les cubes du groupe. Cependant, un mot de passe défini pour un membre du groupe de cubes remplace le mot de passe défini au niveau racine du groupe si la propriété Password de ce cube

particulier est modifiée. Le code d'accès DataSource est, par ailleurs, importée avec les fichiers .iqd ou lorsqu'une source de données est configurée pour inviter à saisir un mot de passe.

Pour utiliser IBM Cognos dans le but de s'authentifier auprès d'un espace-noms externe, l'objet de code d'accès doit être configuré avec l'ID utilisateur, le mot de passe et l'espace-noms associé. L'objet de code d'accès doit être créé en premier.

S'il s'agit d'un objet de code d'accès, la propriété Password spécifie le mot de passe requis pour accéder automatiquement à une base de données. Cet accès exige aussi la définition de la propriété UserID.

Une connexion automatique à la base de données peut être configurée en stockant le mot de passe dans l'objet de code d'accès ainsi que le nom de la base de données logique.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Ecriture

## Exemples

```
signon = model.Signons.Add()signon.Password = "sa"
```

---

## Propriété PatFile

La propriété PatFile définit ou retourne l'emplacement du fichier de modèle cogtr\_*langue*.pat, tel que cogtr\_en.pat, pour l'environnement local du produit associé.

Ce fichier est utilisé pour la conception automatique de modèles utilisant la reconnaissance de structures. Il est installé avec IBM Cognos Transformer dans le répertoire *emplacement\_installation/bin*.

## Syntaxe

*Application* .PatFile

## Application

Application Object

## Explication

Cette propriété définit l'emplacement où Transformer peut rechercher le fichier de modèle cogtr\_*langue*.pat, tel que cogtr\_en.pat, pour l'environnement local du produit associé.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété Path

La propriété Path retourne l'emplacement d'un fichier modèle pour un objet Model. Pour un objet Report ou Pack, elle renvoie le chemin vers le rapport ou le pack dans IBM Cognos Connection.

## Syntaxe

*object* .Path

## Application

Objet Model

Objet Package

Objet Report

## Explication

Utilisez cette propriété pour retourner l'emplacement d'un fichier modèle ou pour consigner les informations relatives à cet emplacement dans un fichier journal. Pour retourner le nom de fichier, utilisez la propriété FileName. Pour retourner le chemin d'accès complet et le nom de fichier, utilisez la propriété FullName.

Pour spécifier l'emplacement dans lequel les modèles doivent être enregistrés, utilisez la propriété ModelsPath.

Cette propriété permet d'extraire ou de définir le chemin vers une source de données de pack ou de rapport. Voici un exemple de syntaxe d'un chemin d'accès à un pack.

```
/content/package[@name='GO Data Warehouse (analysis)']
```

## Type

Chaîne

## Accès

Lecture pour le modèle

Lecture/écriture pour le pack ou le rapport

## Exemples

```
new_package = model.Packages.Add()new_package.Path = "/content/package[@name='GO Data Warehouse (analysis)']"
```

---

## Propriété PopulateByDataSource

La propriété PopulateByDataSource définit ou retourne une valeur indiquant si le taux de conversion est obtenu à partir d'une source de données ou s'il est indiqué dans Transformer.

### Syntaxe

*CurrencyRate* .PopulateByDataSource

### Application

Objet CurrencyRate

### Explication

La valeur True indique que les taux sont définis dans une source de données externe. La valeur False indique que les taux de conversion sont définis dans Transformer.

Paramètre par défaut : False

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRate  
= objCurrencyRecord.CurrencyRates(intX)If objCurrencyRate.PopulateByDataSource  
= False Then
```

---

## Propriété Position

La propriété Position définit ou retourne la position de la colonne dans la source de données.

### Syntaxe

*Column* .Position

### Application

Objet Column

### Explication

Pour toutes les zones sauf celles issues de sources de données de texte à zones fixes, cette position correspond à la position ordinale d'une colonne dans la source de données. Les positions de colonne de la source commencent à 1.

Si votre source de données est un fichier de texte à zones fixes, la propriété Position correspond à l'octet de départ d'une colonne. Le premier octet d'un

enregistrement est l'octet numéro 1. Une valeur de position zéro (0) signifie qu'il manque des valeurs dans la colonne. Si, par exemple, la première zone contient 30 octets (un octet par caractère), la position de décalage du premier octet de la deuxième zone est 31.

### Type

Long

### Accès

Lecture/écriture

---

## Propriété PowerCubesPath

La propriété PowerCubesPath définit ou retourne l'emplacement dans lequel Transformer crée des fichiers de PowerCube (.mdc).

### Syntaxe

*Application* .PowerCubesPath

### Application

Objet Application

### Explication

Cette propriété permet de spécifier l'emplacement d'un PowerCube. Utilisez la propriété MDCFile pour spécifier un nom de fichier.

Lorsqu'aucun répertoire n'est spécifié, Transformer recherche les emplacements suivants (dans l'ordre) :

- celui spécifié par la propriété ModelsPath,
- le répertoire d'installation de PowerPlay

### Type

Chaîne

### Accès

Lecture/écriture

### Exemples

```
objTransApp.PowerCubesPath = strStartLocation
```

---

## Propriété PowerPlayPath

La propriété PowerPlayPath définit ou retourne l'emplacement du fichier exécutable PowerPlay.exe. Ce fichier exécutable est installé en même temps que l'application PowerPlay.

## Syntaxe

*Application* .PowerPlayPath

## Application

Objet Application

## Explication

Cette propriété permet d'indiquer l'emplacement où Transformer peut trouver le fichier PowerPlay.exe.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété Precision

La propriété Precision définit ou retourne le nombre de décimales des mesures utilisées dans des calculs.

## Syntaxe

*Measure* .Precision

## Application

Objet Measure

## Explication

Cette propriété permet de spécifier le nombre de décimales que PowerPlay utilise pour calculer les valeurs des cumuls et des catégories calculées. Cette propriété est disponible uniquement avec les mesures calculées ou lorsque la valeur de la propriété StorageType de l'objet Measure est trStorageBigFloat.

Quand la valeur de la propriété Precision est inférieure à celle de la propriété OutputScale, PowerPlay l'arrondit à la décimale supérieure.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objMeasure.CategoryCountLevel = objLevel objMeasure.Precision = 0
```

---

## Propriété PromptForPassword

La propriété PromptForPassword indique si les utilisateurs sont toujours invités à saisir un mot de passe lorsqu'ils utilisent Transformer en mode IU.

### Syntaxe

```
Signon .PromptForPassword
```

### Application

Objet de code d'accès

### Explication

Utilisez cette propriété pour indiquer si les utilisateurs sont toujours invités à saisir un mot de passe lorsqu'ils utilisent Transformer en mode IU. Cette propriété s'applique uniquement au type de code d'accès DataSource.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
signon = model.Signons.Add()signon.PromptForPassword = False
```

---

## Propriété Prompts

La propriété Prompts renvoie la collection d'objets Prompts associés à une requête.

### Syntaxe

```
Query .Prompts
```

### Application

Objet Query

### Explication

Cette propriété permet d'effectuer une itération dans les invites d'un objet Query, ainsi que d'ajouter, de modifier ou de supprimer une invite.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Demandes

## Accès

Lecture

## Exemples

```
new_report = model.Reports.Add()new_query = new_report.Queries.Add()new_prompt  
= new_query.Prompts.Add()
```

---

## Propriété PromptValueType

La propriété PromptValueType définit ou renvoie le type de propriété.

## Syntaxe

*Prompt* .PromptValueType

## Application

Objet Prompt

## Explication

Cette propriété doit être affectée avant de définir la propriété Value de l'objet Prompt.

PromptValueType utilise les valeurs de xtrPromptValueType.

## Type

Constante - xtrPromptValueType

## Accès

Lecture/écriture

## Exemples

```
new_report = model.Reports.Add()new_query  
= new_report.Queries.Add()new_prompt.PromptValueType  
= xtrPromptValueType.trSingleValuePrompt
```

---

## Propriété QualifiedName

La propriété QualifiedName renvoie le nom complet du niveau.

## Syntaxe

*object* .QualifiedName



## Application

Objet DateLevel

Objet de niveau

## Explication

Cette propriété permet de déterminer la dimension à laquelle appartient un niveau.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

---

## Propriété QuarterType

La propriété QuarterType définit le mode de calcul du niveau trimestre d'une dimension de temps.

## Syntaxe

*DateWizard* .QuarterType

## Application

Objet DateWizard

## Explication

Cette propriété permet de déterminer si un trimestre fait partie d'une année calendaire ou d'une année lunaire.

La propriété QuarterType utilise les valeurs de la constante xtrSpecialFunction.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrSpecialFunction

## Accès

Écriture

## Exemples

```
objModel.QuarterType = xtrSpecialFunction.trSpecialFunctionQuarter
```

---

## Propriété Queries

La propriété Queries renvoie la collection d'objets Query associée à un objet Package ou Report.

### Syntaxe

*object* .Queries

### Application

Objet Package

Objet Report

### Explication

Cette propriété permet d'effectuer une itération dans les requêtes d'un objet Package ou Report, de modifier ou de supprimer des requêtes.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Requêtes

### Accès

Lecture

### Exemples

```
new_package = model.Packages.Add()new_query = new_package.Queries.Add()
```

---

## Propriété QyPath

La propriété QyPath renvoie l'emplacement d'un modèle en attente.

### Syntaxe

*SuspendedModel* .QyPath

### Application

Objet SuspendedModel

### Explication

Au fur et à mesure que vous travaillez sur un modèle, Transformer définit des points de contrôle dans un fichier temporaire qu'il crée lui-même. Ces fichiers temporaires sont supprimés lorsque le modèle se ferme normalement. La présence de l'un d'eux indique que Transformer s'est arrêté de manière inattendue.

Quand un modèle ne se ferme pas correctement, notamment lors d'une défaillance du système ou d'une panne secteur, Transformer conserve des informations

relatives au modèle en attente jusqu'au dernier point de contrôle. Le modèle en attente est enregistré avec une extension de fichier .qy? (la valeur ? dépend de la version de Transformer).

Utilisez la propriété `SuspendedModels` pour retourner une collection de modèles en attente. Utilisez la propriété `IsBad` pour vérifier si un modèle de la collection est corrompu ou récupérable.

## Type

Chaîne

## Accès

Lecture

## Exemples

```
objSuspendedModel = objTransApp.SuspendedModels(intX)MsgBox("Model :  
" + objSuspendedModel.ModelName + "_", located at " + objSuspendedModel.QyPath  
+ " is corrupt")
```

---

## Propriété Rate

La propriété `Rate` définit et renvoie le taux de conversion monétaire.

## Syntaxe

*CurrencyRate* .Rate

## Application

Objet `CurrencyRate`

## Explication

Transformer crée une collection d'objets `CurrencyRate` pour chaque enregistrement ajouté à une table des devises. La taille de la collection dépend du niveau de détail et de la plage de dates. Un intervalle de deux ans et un niveau de détail défini sur le mois, par exemple, requiert 24 taux de conversion.

Si l'enregistrement est créé manuellement, utilisez cette propriété pour définir et gérer le taux au sein du modèle.

Inversement, si des enregistrements sont chargés à partir d'une source de données externe, les taux sont définis et gérés dans la source de données. Dans ce cas, utilisez la propriété `Rate` pour retourner la valeur spécifiée dans la source de données.

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Double

## Accès

Lecture/écriture

## Exemples

```
objCurrencyRecord = objModel.CurrencyRecords.Add()objCurrencyRate  
= objCurrencyRecord.CurrencyRates(intX)objCurrencyRate.Rate = 1.54
```

---

## Propriété RefName

La propriété RefName définit ou retourne le nom du filtre dans la source de données.

## Syntaxe

*Filter* .RefName

## Application

Objet Filter

## Explication

Utilisez cette propriété pour établir des correspondances croisées entre le filtre de votre modèle et celui de la source de données d'origine.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
new_package = model.Packages.Add()new_query = new_package.Queries.Add()new_filter  
= new_query.Filters.Add()new_filter.RefName = "[go_data_warehouse].[2004]"
```

---

## Propriété RefreshDescription

La propriété RefreshDescription définit ou renvoie une valeur indiquant si les descriptions sont mises à jour.

## Syntaxe

*object* .RefreshDescription

## Application

Objet DateLevel

Objet de niveau

## Explication

Cette propriété permet de mettre à jour la propriété Description des catégories chaque fois que Transformer génère des catégories. Le niveau doit inclure un objet Association dont la propriété AssociationRole a pour valeur la constante trAssociationDescription.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee") objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.RefreshDescription
= True
```

---

## Propriété RefreshLabel

La propriété RefreshLabel définit ou renvoie une valeur indiquant si les libellés sont mis à jour.

## Syntaxe

*object* .RefreshLabel

## Application

Objet DateLevel

Objet de niveau

## Explication

Cette propriété permet de mettre à jour la propriété Label des catégories chaque fois que Transformer génère des catégories. Le niveau doit inclure un objet Association dont la propriété AssociationRole a pour valeur la constante trAssociationLabel.

Valeur par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim
= objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.RefreshLabel
= True
```

---

## Propriété RefreshShortName

La propriété RefreshShortName définit ou renvoie une valeur indiquant si les noms abrégés sont mis à jour.

### Syntaxe

*object* .RefreshShortName

### Application

Objet DateLevel

Objet de niveau

### Explication

Cette propriété permet de mettre à jour la propriété ShortName des catégories chaque fois que Transformer génère des catégories. Le niveau doit inclure un objet Association dont la propriété AssociationRole a pour valeur la constante trAssociationShortName.

Paramètre par défaut : False

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objLocationsDim
= objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.RefreshShortName
= True
```

---

## Propriété RegularRollup

La propriété RegularRollup définit ou retourne le type de fonction de cumul ordinaire en vigueur.

### Syntaxe

*Measure* .RegularRollup

## Application

Objet Measure

## Explication

Cette propriété permet de modifier la manière dont les valeurs d'une mesure sont cumulées.

Un cumul récapitule les valeurs de chaque catégorie d'un niveau en effectuant un calcul sur toutes les catégories enfants. Par défaut, PowerPlay utilise la fonction Sum dans le cumul des catégories. Vous pouvez remplacer cette valeur par défaut par minimum, maximum, moyenne, nombre, nombre complet, quelconque ou externe à l'aide des constantes xtrRollup.

Transformer consolide les enregistrements qui contiennent des valeurs de non-mesure en double, c'est-à-dire celles dont les noms de catégorie sont identiques.

N'oubliez pas de définir la propriété RegularWeight quand la propriété RegularRollup a pour valeur trRollupAverage.

Lorsque vous utilisez les propriétés RegularRollup et TimeStateRollup, PowerPlay effectue d'abord le cumul ordinaire, puis le cumul d'état temporel.

Les propriétés CanAllocateMeasure, Consolidate, DuplicateRollup, TimeStateRollup et RollupTiming influencent aussi la manière et le moment auquel les mesures sont cumulées.

Cette propriété utilise les valeurs de la constante xtrRollup.

## Type

Constante - xtrRollup

## Accès

Lecture/écriture

## Exemples

```
objMeasures
= objModel.MeasurecurrentMeasure
= objMeasures("Revenue").currentMeasure.RegularRollup
= xtrRollup.trRollupAverage
```

---

## Propriété RegularWeight

La propriété RegularWeight définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.

## Syntaxe

*Measure* .RegularWeight

## Application

Objet Measure

## Explication

Utilisez la propriété RegularWeight pour obtenir une moyenne pondérée de la mesure de cumul plutôt qu'une moyenne réelle.

La moyenne pondérée de la mesure A, qui utilise la mesure B comme pondération, est calculée à l'aide d'une équation similaire à celle-ci :

$$\sum(A_i * B_i) / \sum B_i$$

Le paramètre xtrRollup de la mesure qui est cumulée doit avoir la valeur trRollupAverage, tandis que celui de la mesure spécifiée comme mesure de pondération doit avoir pour valeur la constante trRollupDefault ou trRollupSum.

Vous ne pouvez pas spécifier de moyenne pondérée à la fois pour des cumuls ordinaires et pour des cumuls d'état temporel.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure  
= objMeasures("Revenue").currentMeasure.RegularWeight  
= objModel.Measures("Quantity").Name
```

---

## Propriété Reports

La propriété Reports renvoie la collection des objets Report associés à un modèle Transformer.

## Syntaxe

*Model* .Reports

## Application

Objet Model

## Discussion

Cette propriété permet d'effectuer une itération dans les rapports d'un objet Model, d'ajouter un nouveau rapport, de modifier ou supprimer un rapport.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



## Type

Objets - Reports

## Accès

Lecture

## Exemples

```
new_report = model.Reports.Add()
```

---

## Propriété ReverseSign

La propriété ReverseSign définit ou retourne une valeur indiquant si PowerPlay inverse le signe d'une mesure.

### Syntaxe

*object* .ReverseSign

### Application

Objet Category

Objet Measure

Objet SpecialCategory

### Explication

Quand vous affectez la valeur True à la propriété ReverseSign, vous inversez le signe des valeurs d'une mesure et des catégories basées sur cette mesure.

Cette propriété permet d'inverser le signe dans les systèmes de rapports financiers dans lesquels des mesures, telles que les dépenses, doivent afficher un signe positif dans certains contextes et négatif dans d'autres.

La propriété ReverseSign doit avoir la valeur True à la fois pour la mesure et pour la catégorie associée afin que PowerPlay affiche les valeurs avec le signe inversé.

**Paramètre par défaut** : False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objMeasure.CategoryCountLevel = objLevelobjMeasure.CategoryCountLevel  
= objLevel objMeasure.ReverseSign = False
```

---

## Propriété Rollup

La propriété Rollup définit ou retourne une valeur indiquant si les valeurs de mesure d'une catégorie spéciale sont cumulées dans la catégorie parent.

### Syntaxe

*SpecialCategory* .Rollup

### Application

Objet SpecialCategory

### Explication

Une catégorie qui possède une valeur cumulée affiche un récapitulatif des valeurs des catégories enfants. Lorsque cette propriété a pour valeur False pour tous les descendants d'une catégorie spéciale, celle-ci n'est utilisée que comme indicateur d'emplacement à l'intérieur de la hiérarchie des catégories spéciales.

Paramètre par défaut : True

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories.Add(xtrObjectType
.trSpecialCategory)objSpecCategory.Rollup
= True
```

---

## Propriété RollupTiming

La propriété RollupTiming définit ou retourne une valeur indiquant à quel moment les calculs des mesures calculées doivent être effectués.

### Syntaxe

*Measure* .RollupTiming

### Application

Objet Measure

### Explication

Les catégories qui possèdent des valeurs cumulées affichent un récapitulatif des valeurs de leurs catégories enfants respectives. Les catégories calculées peuvent effectuer des calculs sur des valeurs de mesure avant ou après le récapitulatif de ces valeurs. Utilisez cette propriété pour spécifier le temps de ces calculs.

Quand une mesure calculée utilise `trTimingBeforeRollup`, vous ne pouvez employer ni le découpage automatique en partitions ni la consolidation des enregistrements. Pour inclure le découpage automatique en partitions et la consolidation dans un cube, utilisez des colonnes calculées plutôt que des mesures calculées. Les colonnes calculées peuvent effectuer les mêmes calculs avant le cumul.

Cette propriété utilise les valeurs de la constante `xtrRollupTiming`.

## Type

Constante - `xtrRollupTiming`

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure  
= objMeasures("Revenue").currentMeasure.RollupTiming  
= xtrRollupTiming.trTimingDefault
```

---

## Propriété `RowsAsSample`

La propriété `RowsAsSample` définit ou retourne le nombre de lignes que la méthode `DoAutoDesign` utilise comme échantillon pour créer un modèle.

## Syntaxe

*Application* .**RowsAsSample**

## Application

Objet `Application`

## Explication

Cette propriété permet de spécifier le nombre de lignes que la méthode `DoAutoDesign` utilise comme échantillon quand elle détermine les relations hiérarchiques possibles des niveaux. Ce nombre ne doit pas être supérieur à celui défini par la propriété `RowsChecked`.

Minimum : 10 lignes. Maximum : 1000 lignes.

**Paramètre par défaut** : 300 lignes

Une exception `COM` est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objTransApp.RowsAsSample = 600
```

---

## Propriété RowsChecked

La propriété RowsChecked définit ou retourne le nombre maximum de lignes que la méthode DoAutoDesign lit à partir de la source de données.

### Syntaxe

*Application* .RowsChecked

### Application

Objet Application

### Explication

La méthode DoAutoDesign vérifie les lignes pour déterminer les colonnes qui correspondent aux sources des niveaux et des mesures.

Spécifiez un nombre élevé de lignes pour augmenter la précision de la méthode DoAutoDesign. Un nombre de lignes réduit, en revanche, permet d'accélérer le processus de création du modèle par Transformer.

Si vous spécifiez la valeur zéro, la méthode DoAutoDesign lit la totalité de la source de données.

**Valeur par défaut :** 600 lignes.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

## Accès

Lecture/écriture

## Exemples

```
objTransApp.RowsChecked = 600
```

---

## Propriété RunningPeriods

La propriété RunningPeriods définit ou retourne le nombre de périodes de date utilisées dans le calcul des totaux cumulatifs.

## Syntaxe

*SpecialCategory* .RunningPeriods

## Application

Objet SpecialCategory

## Explication

Utilisez cette propriété avec les propriétés TargetLevel, ContextLevel, TargetOffset et ContextOffset pour définir une catégorie spéciale assurant le suivi des mesures pour des périodes de temps spécifiques par rapport à la date en cours.

Pour utiliser cette propriété, la propriété Aggregate doit avoir la valeur trAggregateRunning ou trAggregateRunningGrouped. Créez, par exemple, une catégorie de total cumulatif qui s'étend sur les six mois qui précèdent le mois en cours, pour cette année et l'année dernière.

```
Aggregate = trAggregateRunningGrouped
ContextLevel = "Year"
ContextOffset = -1
TargetLevel = "Month"
TargetOffset = 0
RunningPeriods = 6
```

## Type

Long

## Accès

Lecture/écriture

## Exemples

```
objSpecCategory
= _objModel.Dimensions("Time").Categories
  .Add(xtrObjectType.trSpecialCategory)objSpecCategory.RunningPeriods
= 5
```

---

## Propriété SecurityObjects

La propriété SecurityObjects renvoie une collection de SecurityObjects.

## Syntaxe

*Object* .SecurityObjects

## Application

Objet CustomView

Objet Namespace

## Explication

Cette propriété permet d'effectuer une itération dans les objets de sécurité d'un espace-noms, d'ajouter de nouveaux objets de sécurité, de modifier ou supprimer un objet de sécurité.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

SecurityObjects

## Accès

Lecture

## Exemples

```
new_namespace = model.Namespaces.Add()securityObject  
= new_namespace.SecurityObjects.Add()
```

---

## Propriété Server

La propriété Server définit ou retourne une valeur indiquant si un cube est traité localement ou sur un serveur.

## Syntaxe

*object* .Server

## Application

Objet Cube

Objet CubeGroup

## Explication

La valeur True signifie que le cube est traité sur un serveur.

Utilisez la propriété ServerModelPath pour spécifier le nom du fichier modèle (.mdl) sur le serveur. La propriété ServerConnect permet de spécifier le nom de connexion du serveur.

**Paramètre par défaut :** False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
securityObject = new_namespace.SecurityObjects.Add()
```

---

## Propriété ServerModelPath

La propriété ServerModelPath définit ou retourne le nom d'un fichier modèle (.mdl) sur un serveur.

### Syntaxe

*Model* .ServerModelPath

### Application

Objet Model

### Discussion

Cette propriété permet de spécifier le nom de fichier. Transformer (sous UNIX) enregistre le modèle dans le répertoire par défaut. Pour enregistrer le fichier modèle dans un autre répertoire, créez un paramètre préférentiel qui spécifie le chemin d'accès complet et le nom de fichier.

Utilisez la propriété ServerConnect pour spécifier la chaîne de connexion.

### Type

Chaîne

### Accès

Lecture/écriture

---

## Propriété ServerPath

La propriété ServerPath définit ou retourne le nom et l'emplacement d'un fichier de la source de données.

### Syntaxe

*object* .ServerPath

### Application

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

### Explication

Vous pouvez spécifier le chemin d'accès complet ou seulement le nom du fichier. Si vous ne spécifiez que le nom de fichier, Transformer recherche un nom de

répertoire dans la propriété DataSourcePath. Si DataSourcePath n'est pas défini, Transformer vérifie le répertoire d'installation de PowerPlay.

Si le modèle Transformer est traité localement, utilisez la propriété LocalPath pour spécifier l'emplacement des sources de données.

### **Type**

Chaîne

### **Accès**

Lecture/écriture

---

## **Propriété ServerQuery**

La propriété ServerQuery définit ou retourne une valeur indiquant si les données sont traitées localement ou sur un serveur.

### **Syntaxe**

*object* .ServerQuery

### **Application**

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

### **Explication**

La valeur True signifie que les données sont traitées sur un serveur. Dans ce cas, utilisez la propriété ServerPath pour spécifier l'emplacement de la source de données.

**Valeur par défaut :** False

### **Type**

Booléen

### **Accès**

Lecture/écriture

---

## **Propriété ServicesBuildNumber**

La propriété ServiceBuildNumber renvoie le numéro de version de Transformer.



## Syntaxe

*Application* .ServicesBuildNumber

## Application

Objet Application

## Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Long

## Accès

Lecture

---

## Propriété ServicesVersionText

La propriété ServicesVersionText renvoie la version de Transformer.

## Syntaxe

*Application* .ServicesVersionText

## Application

Objet Application

## Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

---

## Propriété SetsCurrentPeriod

La propriété SetsCurrentPeriod définit ou retourne une valeur indiquant si Transformer recherche la date de la période en cours dans une source de données.

## Syntaxe

*object* .SetsCurrentPeriod

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

Objet Query

## Explication

Quand la valeur est True, Transformer utilise la source de données associée pour déterminer la date de période en cours, c'est-à-dire la date la plus récente de la colonne de date.

Transformer utilise la date de période en cours pour évaluer des catégories de date relative. Par exemple, si la date en cours est 31/08/2001, la catégorie du mois précédent affiche la valeur Juillet.

Si le modèle contient plusieurs sources de données, sélectionnez celles dans lesquelles Transformer recherche la date de période en cours. Quand cette propriété n'est pas spécifiée, Transformer vérifie toutes les sources de données et sélectionne la date la plus récente.

Affectez à cette propriété la valeur False dans une source de données utilisée pour remplir une table des devises. Assurez-vous aussi que le nom de la colonne de date diffère de celui utilisé par la dimension de temps.

**Valeur par défaut :** True

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources
  .Add(xtrObjectType.trFlatFileDataSource)objDataSource.SetsCurrentPeriod
= True
```

---

## Propriété ShortName

La propriété ShortName définit ou retourne le nom abrégé d'un objet.

## Syntaxe

*object* .ShortName

## Application

Objet Category

Objet DateDrillDown

Objet DrillDown

Objet Measure

Objet SpecialCategory

## Explication

Cette propriété permet d'afficher un nom plus significatif pour un objet dans PowerPlay.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objMeasures = objModel.MeasurescurrentMeasure  
= objMeasures("Revenue").currentMeasure.ShortName  
= "Revenue"
```

---

## Propriété Signon

La propriété Signon définit ou renvoie l'objet de code d'accès associé au cube, au groupe de cubes ou à une connexion à une source de données de pack (code d'accès IBM Cognos).

## Syntaxe

*object* .Signon

## Application

Objet Cube

Objet CubeGroup

Objet PackageDatasourceConnection

## Explication

Quand vous ajoutez un fichier de définition de requête d'Impromptu (.iqd) à votre modèle, Transformer ajoute automatiquement un objet de code d'accès. Cet objet contient le nom de base de données logique et éventuellement des informations relatives à l'ID utilisateur et au mot de passe.

Pour utiliser le code d'accès PackageDatasourceConnection, un objet de code d'accès doit être ajouté afin de permettre l'authentification vers un espace-noms externe. Les utilisateurs peuvent alors créer des cubes en mode différé. Pour permettre à Transformer d'utiliser automatiquement le code d'accès IBM Cognos, la propriété AutoLogon de l'objet de code d'accès doit être activée. Ce code d'accès gère l'ID utilisateur, le mot de passe et l'espace-noms associé.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Code d'accès (objets Cube et Cube Group)

Chaîne (PackageDatasourceConnection)

## Accès

Lecture/écriture

## Exemples

```
package = model.Packages.Add()connection  
= package.PackageDatasourceConnections.Add()connection.Signon  
= "great_outdoors_warehouse"
```

---

## Propriété SignOnNamespace

La propriété SignOnNameSpace contient l'espace-noms sécurisé associé au code d'accès IBM Cognos.

## Syntaxe

*Signon* .SignonNamespace

## Application

Objet de code d'accès

## Explication

La propriété SignOnNamespace s'applique uniquement au type de code d'accès IBM Cognos.

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
signon = model.Signons.Add()signon.SignOnNamespace =  
"Cognos"
```

---

## Propriété Signons

La propriété Signons renvoie la collection Signons pour un modèle.

### Syntaxe

*Model* .Signons

### Application

Objet Model

### Explication

Avant de pouvoir ajouter ou modifier des objets de code d'accès, vous devez retourner la collection Signons.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet

### Accès

Lecture

### Exemples

```
signon = model.Signons.Add()
```

---

## Propriété SignonType

La propriété SignonType définit ou retourne le type de code d'accès xtrSignonType.

### Syntaxe

*Signon* .SignonType

### Application

Objet de code d'accès

### Explication

trDataSourceSignon est utilisé pour un code d'accès à une source de données et pour xtrSignonType. trCognosSignon est utilisé pour un code d'accès IBM Cognos.

### Type

Constante - xtrSignonType

## Accès

Lecture/écriture

## Exemples

```
signon = model.Signons.Add()signon.SignonType = xtrSignonType.trDataSourceSignon
```

---

## Propriété Size

La propriété Size définit ou retourne la taille d'une colonne ou d'un fichier modèle (.mdl).

### Syntaxe

*object* .Size

### Application

Objet Column

Objet Model

### Explication

Dans le cas d'un objet Column, cette propriété définit ou retourne la taille de la colonne en octets pour certaines sources de données. La taille de colonne est appliquée uniquement lorsque la propriété SourceType associée à la source de données possède la valeur trPowerHousePortable, trFixedAscii ou trFixedAsciiNoCRLF.

Dans le cas d'un objet Model, cette propriété renvoie la taille du fichier telle qu'elle apparaît dans un dossier Windows ou dans l'Explorateur Windows.

### Type

Long (Column)

Chaîne (Model)

### Accès

Lecture/Ecriture (Column)

Lecture (Model)

### Exemples

```
MsgBox(objModel.FileName & " " & .Size &  
" " & .Time)
```

---

## Propriété SortComparisonRule

La propriété SortComparisonRule définit ou retourne la règle de comparaison de texte utilisée par Transformer pour trier des noms de catégorie.

## Syntaxe

*Application* .SortComparisonRule

## Application

Objet Application

## Explication

Vous pouvez choisir la règle de comparaison déterminée par les paramètres régionaux dans le Panneau de configuration de Windows ou la règle de comparaison interne de Transformer, qui compare chaque octet de la représentation binaire de chaque chaîne.

Si vous modifiez la valeur de cette propriété pour un modèle existant, la nouvelle règle ne s'applique qu'aux nouvelles catégories. Les catégories existantes ne sont pas reclassées. Il se peut donc que le classement des catégories existantes ne soit pas cohérent avec celui des nouvelles catégories.

Pour que Transformer reclasse des catégories dans un niveau, vous devez supprimer l'objet Association qui contient les informations de classement, puis en créer un nouveau.

Pour trier des catégories dans un niveau, créez un objet Association pour ce niveau et affectez à la propriété AssociationRole la valeur trAssociationOrderBy.

Cette propriété utilise les valeurs de la constante xtrPreferences.

## Type

Constante - xtrPreferences

## Accès

Lecture/écriture

## Exemples

```
objTransApp.SortComparisonRule = xtrPreferences.trSortIgnoreControlPanel
```

---

## Propriété SourceType

La propriété SourceType définit ou retourne le type de fichier de données.

## Syntaxe

*object* .SourceType

## Application

Objet CrossTabDataSource

Objet DataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

Chaque objet source de données d'un modèle référence un type de fichier de données. Utilisez cette propriété avec les valeurs de la constante `xtrSourceType` pour définir la source appropriée.

Quand vous ajoutez une source de données à une collection, le paramètre `Type` de la méthode `Add` utilise une constante de la liste des valeurs de `xtrObjectType`. La méthode `Add` utilise par exemple le paramètre `trCrossTabDataSource` pour spécifier un fichier de feuille de calcul. Vous utilisez ensuite la propriété `SourceType` pour spécifier la manière dont Transformer lit le fichier. Dans cet exemple, la propriété `SourceType` détermine si le fichier de feuille de calcul est un tableau ou un fichier de base de données.

## Type

Constante - `xtrSourceType`

## Accès

Lecture/écriture

## Exemples

```
objDataSource
= objModel.DataSources
  .Add(xtrObjectType.trFlatFileDataSource)objDataSource.SourceType
= xtrSourceType.trFlatFileColumnNames
```

---

## Propriété SpecialCategoryCount

La propriété `SpecialCategoryCount` retourne le nombre de changements de niveaux, de racines et de catégories spéciales dans une dimension.

## Syntaxe

*object* .`SpecialCategoryCount`

## Application

Objet `DateDimension`

Objet `Dimension`

## Explication

Cette propriété ne compte que les changements de niveaux, les racines et les catégories spéciales. Pour compter des objets `Category`, utilisez la propriété `CategoryCount`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.



## Type

Long

## Accès

Lecture

---

## Propriété SQLExpression

La propriété SQLExpression renvoie une expression SQL.

### Syntaxe

*IqdDataSource* .SQLExpression

### Application

Objet IqdDataSource

### Explication

Utilisez cette propriété pour examiner l'expression SQL sur laquelle est basé un fichier de définition de requête d'Impromptu (.iqd).

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété Status

La propriété Status renvoie le statut de création précédent d'un cube.

### Syntaxe

*object* .Status

### Application

Objet ChildCube

Objet Cube

Objet CubeGroup

## Explication

Les informations de statut permettent de qualifier un cube pour la production. Vous pouvez, par exemple, affecter la valeur False à la propriété CubeCreation pour tous les cubes pour lesquels des avertissements ont été émis.

La propriété Status utilise les valeurs de la constante xtrCubeStatus.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrCubeStatus

## Accès

Lecture

---

## Propriété StorageType

La propriété StorageType définit ou retourne la taille d'un type de données numérique.

## Syntaxe

*object* .StorageType

## Application

Objet Column

Objet Measure

## Explication

Cette propriété permet de spécifier la manière dont Transformer stocke des données de mesure à l'intérieur des fichiers de travail utilisés pour créer des cubes. Spécifiez une constante de la liste des valeurs de xtrStorage suffisamment élevée pour contenir les valeurs sources.

Assurez-vous que vous spécifiez une valeur suffisamment élevée pour pouvoir traiter les valeurs de cumul et la consolidation des enregistrements.

## Type

Constante - xtrStorage

## Accès

Lecture/écriture

## Exemples

```
objMeasure.StorageType = xtrStorage.trStorageDefault
```

---

## Propriété StreamExtractAllowed

La propriété StreamExtractAllowed définit ou renvoie une valeur indiquant si l'extraction de flux est autorisée. L'extraction de flux s'applique uniquement à une source de données SAP BW.

### Syntaxe

*Query* .StreamExtractAllowed

### Application

Objet Query

### Explication

Cette propriété est ignorée pour les sources de données autres que SAP BW.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Booléen

### Accès

Lecture/écriture

---

## Propriété StreamExtractSize

La propriété StreamExtractSize définit ou retourne la taille, en mégaoctets, du tampon utilisé pour transférer des données depuis SAP quand StreamExtract est défini sur True.

### Syntaxe

*Query* .StreamExtractSize

### Application

Objet Query

### Explication

Cette propriété sera définie sur 0 (et ignorée) pour les requêtes de structure.

Elle sera définie sur 10 pour une requête de transaction construite pour extraire des données de mesure.

Quand cette valeur est différente de zéro (0) et que toutes les colonnes ont l'origine correcte, Stream Extract est utilisé pour lire les données.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Entier

## Accès

Lecture/écriture

---

## Propriété SummaryLevel

La propriété SummaryLevel définit ou retourne le niveau à utiliser pour récapituler des catégories externes dans un groupe de cubes.

## Syntaxe

*CubeGroup* .SummaryLevel

## Application

Objet CubeGroup

## Explication

Utilisez cette propriété pour récapituler les catégories des autres cubes du groupe de cubes.

Les cubes enfants d'un groupe de cubes représentent chacun une catégorie. Du point de vue d'un cube enfant unique, les catégories externes correspondent à toutes les catégories de la dimension autres que celle référencée par ce cube enfant et les catégories descendantes de la catégorie. Dans PowerPlay, les utilisateurs peuvent consulter des données d'autres cubes du groupe jusqu'au niveau spécifié par cette propriété.

Si aucune valeur n'est définie, Transformer exclut toutes les catégories externes de chaque cube du groupe.

## Type

Objet

## Accès

Lecture/écriture

## Exemples

```
objCubesByRegion  
= objModel.Cubes.Add(xtrObjectType.trCubeGroup)objCubesByRegion.SummaryLevel  
= objRegionsDrill.Levels("Sales region")
```

---

## Propriété SuppressNull

La propriété SuppressNull définit ou renvoie l'option de suppression des valeurs nulles utilisée pour les sources de données SAP BW.

## Syntaxe

*Query* .SuppressNull

## Application

Objet Query

## Explication

Cette option s'applique uniquement aux sources de données SAP BW.

Certaines requêtes peuvent être volumineuses car les valeurs nulles ne sont pas supprimées par filtrage. La suppression des valeurs nulles permet de supprimer une ligne ou une colonne dont toutes les valeurs sont nulles (vides). Cette opération est effectuée par SAP BW. Elle réduit la quantité de données transférées vers Transformer et améliore les performances.

L'application d'une suppression des valeurs nulles à une source de données permet de contrôler la suppression des valeurs nulles lors de l'extraction des données. Cette opération est différente de l'application d'une suppression des valeurs nulles lors de la publication de packs, en ce sens que cette dernière contrôle uniquement l'affichage des valeurs nulles.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Entier

## Accès

Lecture/écriture

---

## Propriété SuspendedModels

La propriété SuspendedModels renvoie une collection d'objets SuspendedModel.

## Syntaxe

*Application* .SuspendedModels

## Application

Objet Application

## Explication

Si un modèle se ferme brutalement, notamment lors d'une panne de système, Transformer ajoute un objet SuspendedModel à la collection SuspendedModels.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Objet - SuspendedModels

## Accès

Lecture

## Exemples

```
objSuspendedModel = objTransApp.SuspendedModels(intX)
```

---

## Propriété TargetLevel

La propriété TargetLevel définit ou retourne le niveau de détail d'une période de date.

## Syntaxe

*SpecialCategory*.TargetLevel

## Application

Objet SpecialCategory

## Explication

Les catégories spéciales effectuent le suivi des mesures d'une période de date spécifique par rapport à la date en cours.

Cette propriété permet de spécifier une période de rapport. Si la période cible est le mois, par exemple, PowerPlay affiche une période mensuelle.

Utilisez les propriétés TargetOffset, ContextLevel, ContextOffset et RunningPeriods pour répondre aux questions relatives à la période cible. Si la période cible est le mois, par exemple, les propriétés répondent aux questions suivantes :

- S'agit-il du mois en cours (TargetOffset) ?
- Est-elle définie par rapport à un trimestre ou une année (ContextLevel) ?
- S'agit-il du trimestre ou de l'année en cours (ContextOffset) ?
- Combien y a-t-il de périodes cible (RunningPeriods) ?

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
objSpecCategory  
= objModel.Dimensions("Time").Categories.Add(xtrObjectType.  
trSpecialCategory)objSpecCategory.TargetLevel  
= "Month"
```

---

## Propriété TargetOffset

La propriété TargetOffset définit ou retourne la position de la période cible par rapport à la période en cours.

### Syntaxe

*SpecialCategory* .TargetOffset

### Application

Objet SpecialCategory

### Explication

Cette propriété permet de spécifier une valeur de décalage, par rapport à la période de date en cours, pour la propriété TargetLevel. Si la propriété TargetLevel a pour valeur le mois et si vous souhaitez une catégorie spéciale pour le mois précédent, par exemple, affectez à la propriété TargetOffset la valeur -1.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Short

### Accès

Lecture/écriture

### Exemples

```
objSpecCategory
= _ objModel.Dimensions("Time").Categories
  .Add(xtrObjectType.trSpecialCategory)objSpecCategory.TargetOffset
= -1
```

---

## Propriété ThousandPoint

La propriété ThousandPoint définit ou retourne le caractère qui sépare les milliers.

### Syntaxe

*Object* .ThousandPoint

### Application

Objet CrossTabDataSource

Objet DbDataSource

Objet FlatFileDataSource

Objet IqdDataSource

## Explication

Par exemple, vous pouvez utiliser cette propriété pour :

- inclure une virgule dans une valeur telle que 1,000,
- inclure un espace dans une valeur telle que 1 000.

Vous ne pouvez pas utiliser de caractère nul pour cette propriété, ce qui signifie que vous devez indiquer un caractère.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété Time

La propriété Time renvoie l'horodateur d'un modèle tel qu'il apparaît dans un dossier Windows ou dans l'Explorateur Windows.

## Syntaxe

*Model* .Time

## Application

Objet Model

## Explication

Utilisez cette propriété pour vérifier la date de la dernière modification d'un modèle.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

## Exemples

```
MsgBox(objModel.FileName & " " & .Size &  
" " & .Time)
```

---

## Propriété TimeArrayColumn

La propriété TimeArrayColumn définit ou retourne le nom de la première colonne d'un tableau de dates.



## Syntaxe

*Column* .TimeArrayColumn

## Application

Objet Column

## Explication

Un tableau de dates comprend quatre ou douze colonnes adjacentes contenant des valeurs de transaction trimestrielles ou mensuelles. Si vous stockez vos données de transaction par mois ou par trimestre, nous vous conseillons de définir les colonnes en tant que membres d'un tableau de dates et non en tant que mesures individuelles.

Chaque valeur de la colonne de dates représente le premier mois d'un exercice fiscal. Chaque colonne de transaction couvre une période de dates telles qu'un mois ou un trimestre. Pour créer des rapports au niveau trimestriel, par exemple, votre source de données peut contenir des lignes similaires à celles-ci :

```
DATE, PRODUCT, Q1, Q2, Q3, Q4  
199901, Product1, 100, 200, 150, 400  
199901, Product2, 1110, 2265, 1995, 4200  
200001, Product1, 110, 210, 160, 420
```

Une fois que vous avez créé une dimension de temps, retournez la première colonne du tableau, puis utilisez la propriété Name de la même colonne pour définir TimeArrayColumn. Vous devez aussi définir les propriétés TimeArrayType et TimeArrayStartMonth. Transformer ajoute automatiquement les autres colonnes au tableau et définit leur classe de données à l'aide de la constante trDataClassArrayMember. Pour terminer, ajoutez uniquement le premier membre du tableau à la collection Measures. Si vous possédez plusieurs tableaux, ajoutez le premier membre de chacun d'eux à la collection Measures.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété TimeArrayStartMonth

La propriété Time ArrayStartMonth définit ou retourne le premier mois de l'exercice fiscal d'un tableau de dates.

## Syntaxe

*Column* .TimeArrayStartMonth

## Application

Objet Column

## Explication

Un tableau de dates comprend quatre ou douze colonnes adjacentes contenant des valeurs de transaction trimestrielles ou mensuelles. Si vous stockez vos données de transaction par mois ou par trimestre, nous vous conseillons de définir les colonnes en tant que membres d'un tableau de dates et non en tant que mesures individuelles.

Cette propriété nomme le premier mois de l'année indépendamment de la période couverte par le tableau.

Vous devez aussi définir les propriétés `TimeArrayColumn` et `TimeArrayType`.

## Type

Long

## Accès

Lecture/écriture

---

## Propriété `TimeArrayType`

La propriété `TimeArrayType` définit ou retourne le type de tableau utilisé pour les valeurs de date.

## Syntaxe

*Column* .`TimeArrayType`

## Application

Objet `Column`

## Explication

Un tableau de dates comprend quatre ou douze colonnes adjacentes contenant des valeurs de transaction trimestrielles ou mensuelles. Si vous stockez vos données de transaction par mois ou par trimestre, nous vous conseillons de définir les colonnes en tant que membres d'un tableau de dates et non en tant que mesures individuelles.

Cette propriété spécifie un tableau de mois ou de trimestres. Vous devez aussi définir les propriétés `TimeArrayColumn` et `TimeArrayStartMonth`.

La propriété `TimeArrayType` utilise les valeurs de la constante `xtrTimeArrayType`.

## Type

Constante - `xtrTimeArrayType`

## Accès

Lecture/écriture

---

## Propriété TimeRank

La propriété TimeRank définit ou retourne le rang relatif des niveaux de date dans une dimension de temps.

### Syntaxe

*DateLevel* .TimeRank

### Application

Objet DateLevel

### Explication

La valeur absolue d'un rang n'est pas importante. Seule la valeur relative à d'autres rangs de valeurs temporelles est importante. Quand vous ajoutez un objet DateLevel à une dimension de temps, affectez à cette propriété une valeur supérieure au niveau de date précédent.

Quand Transformer définit des rangs, il affecte une valeur supérieure de 10 unités au rang du niveau précédent ou bien une valeur intermédiaire entre le niveau précédent et les niveaux suivants, selon la valeur la plus petite.

Des dimensions de temps possédant plusieurs branches de niveaux peuvent créer des niveaux dont le rang relatif est ambigu. Transformer suppose que deux niveaux d'une même dimension sont égaux (c'est-à-dire, qu'ils représentent le même type de période) s'ils ont le même rang. Cette hypothèse est la même si les noms ou les colonnes source des niveaux sont différents.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

### Accès

Lecture/écriture

---

## Propriété TimeStamp

La propriété TimeStamp définit ou renvoie l'horodatage d'un objet Package ou Report.

### Titre

*objet* .TimeStamp

### Application

Objet Package

## Explication

Cette propriété permet de fournir ou d'obtenir un horodatage pour un pack ou un rapport. Voici un exemple de l'horodatage d'un pack.

```
/content/folder/[@name='Oracle']/package[@name='oracle_gosales']  
/model[@name='2008-04-24T17:46:18.796Z']
```

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
new_package = model.Packages.Add()new_package.TimeStamp  
= "/content/package[@name='GO Data Warehouse (analysis)']" _  
& "/model[@name=' ' & timestamp & '']"
```

---

## Propriété TimeStateRollup

La propriété TimeStateRollup définit ou retourne la période de date utilisée pour les cumuls d'état temporel.

## Syntaxe

*Measure* .TimeStateRollup

## Application

Objet Measure

## Explication

Un cumul récapitule des valeurs de mesure utilisées par des catégories. La propriété TimeStateRollup définit la manière dont Transformer représente l'état d'une mesure à des moments spécifiques.

Affectez d'abord à la propriété DateDegreeofDetail le niveau de détail le plus bas. Par exemple, si votre fichier source contient des transactions quotidiennes, le degré de détail doit être le jour. Utilisez ensuite une constante de la liste des valeurs de xtrTimeRollup pour définir la propriété TimeStateRollup.

Lorsque vous définissez à la fois les propriétés RegularRollup et TimeStateRollup, PowerPlay effectue d'abord le cumul ordinaire, puis le cumul d'état temporel. Le cumul ordinaire n'est toutefois pas appliqué à la dimension de temps.

N'oubliez pas de définir la propriété TimeStateWeight quand la propriété TimeStateRollup a pour valeur trTimeRollupAverage.

Les valeurs des propriétés CanAllocateMeasure, Consolidate, DuplicateRollup, RollupTiming et RegularRollup influencent aussi la manière et le moment auquel les mesures sont cumulées.

## Type

Constante - xtrTimeRollup

## Accès

Lecture/écriture

---

## Propriété TimeStateWeight

### Description

La propriété TimeStateWeight définit ou retourne le nom d'une mesure utilisée dans un calcul de moyenne pondérée.

### Syntaxe

*Measure* .TimeStateWeight

### Application

Objet Measure

### Explication

Utilisez la propriété TimeStateWeight pour obtenir une moyenne pondérée de la mesure de cumul plutôt qu'une moyenne réelle. Si la mesure A utilise la mesure B pour obtenir une moyenne pondérée, par exemple, l'équation du calcul ressemble à celle-ci :

$$\sum(A_i * B_i) / \sum B_i$$

La propriété TimeStateRollup de la mesure récapitulée doit avoir la valeur trTimeRollupAverage tandis que la propriété RegularRollup de la mesure nommée comme mesure de pondération doit avoir la valeur trRollupDefault ou trRollupSum.

Vous ne pouvez pas spécifier de moyenne pondérée à la fois pour des cumuls ordinaires et pour des cumuls d'état temporel.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété ToDateLevel

La propriété ToDateLevel définit ou retourne la période de date utilisée pour les totaux jusqu'à ce jour.

## Syntaxe

*SpecialCategory* .ToDateLevel

## Application

Objet SpecialCategory

## Explication

Les catégories spéciales effectuent le suivi des mesures d'une période de date spécifique par rapport à la date en cours.

La propriété ToDateLevel permet de spécifier une période de rapport. Si la propriété ToDateLevel a pour valeur le trimestre, par exemple, PowerPlay affiche une période trimestrielle. Pour employer cette propriété, vous devez affecter la valeur trAggregateToDate ou trAggregateToDateGrouped à la propriété Aggregate.

Utilisez les propriétés ContextLevel et ContextOffset pour répondre aux questions relatives à la période de niveau jusqu'à ce jour. Si, dans l'exemple ci-dessus, la valeur de niveau jusqu'à ce jour est le trimestre, les propriétés répondent aux questions suivantes :

- Est-elle définie par rapport à un trimestre ou une année (ContextLevel) ?
- S'agit-il du trimestre ou de l'année en cours (ContextOffset) ?

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété TransdaPath

La propriété TransdaPath définit ou retourne l'emplacement du fichier exécutable transda.exe. Ce fichier exécutable est installé en même temps que l'application Transformer. Il est placé dans *répertoire\_installation/CS7Gateways/bin* où *répertoire\_installation* peut correspondre au répertoire suivant :

## Syntaxe

c:\Program Files\Cognos\  
*Application* .TransdaPath

## Application

Objet Application

## Explication

Cette propriété permet d'indiquer l'emplacement où Transformer peut trouver le fichier transda.exe.

## Type

Chaîne

## Accès

Lecture/écriture

---

## Propriété TransformerSignon

La propriété TransformerSignon définit ou renvoie l'objet de code d'accès Transformer associé à une connexion de source de données de pack (code d'accès IBM Cognos).

## Syntaxe

*PackageDatasourceConnection* .TransformerSignon

## Application

Objet PackageDatasourceConnection

## Explication

Pour utiliser le code d'accès PackageDatasourceConnection, un objet de code d'accès doit être ajouté afin de permettre l'authentification vers un espace-noms externe. Les utilisateurs peuvent alors créer des cubes en mode différé. Pour permettre à Transformer d'utiliser automatiquement le code d'accès IBM Cognos, la propriété AutoLogon de l'objet de code d'accès doit être activée. Ce code d'accès gère l'ID utilisateur, le mot de passe et l'espace-noms associé. Cette valeur est utilisée à la place de la propriété Signon si la valeur de la propriété AlwaysUseTransformerSignon est True.

## Type

Chaîne (PackageDatasourceConnection)

## Accès

Lecture/écriture

## Exemples

```
package
= model.Packages.Add()connection
= package.PackageDatasourceConnections.Add()connection.TransformerSignon
= "great_outdoors_warehouse"
```

---

## Propriété Type

La propriété Type renvoie un type d'objet.

## Syntaxe

*objet* .Type

## **Application**

«Objet Application», à la page 45

Objet Association

Objet CalculationDefinition

Objet Category

Objet CategorySet

Objet ChildCube

Objet Column

Objet CrossTabDataSource

Objet Cube

Objet CubeGroup

Objet CurrencyRate

Objet CurrencyRecord

Objet CurrencyTable

Objet CustomView

Objet DataSource

Objet DateDimension

Objet DateDrillDown

Objet DateLevel

Objet DateWizard

Objet DbDataSource

Objet Dimension

Objet DrillDown

Objet DrillThroughTarget

Objet FlatFileDataSource

Objet IqdDataSource

Objet de niveau

Objet Measure



Objet Model

Objet Name

Objet Package

Objet Prompt

«Objet Report», à la page 113

Objet SecurityObject

Objet de code d'accès

Objet SpecialCategory

Objet SuspendedModel

Objet View

## Explication

Cette propriété permet de déterminer le type d'objet. Cette valeur de retour s'avère utile quand vous récupérez un objet d'une collection qui peut contenir plusieurs types d'objet. Un objet retourné à partir d'une collection Dimensions, par exemple, peut-être un objet Dimension ou DateDimension.

La valeur de la propriété Type de chaque objet dans Transformer correspond à une constante de xtrObjectType, sauf pour l'objet DataSource. Dans ce cas, la valeur de la propriété Type est définie par l'objet de la collection DataSources que représente actuellement l'objet DataSource.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrObjectType

## Accès

Lecture

## Exemples

```
new_namespace
= model.Namespaces.Add()securityObject
= new_namespace.SecurityObjects.Add()securityObject.Type
= xtrSecurityType.trSecurityType_Role
```

---

## Propriété Unique

La propriété Unique définit ou retourne une valeur indiquant si chaque catégorie du niveau peut être identifiée à l'aide d'une valeur source unique.

## Syntaxe

*object* .Unique

## Application

Objet DateLevel

Objet de niveau

## Explication

Vous spécifiez qu'un niveau est unique quand il correspond au niveau de convergence des branches de niveaux secondaires ou quand le modèle contient plusieurs sources de données.

Quand cette propriété a la valeur True, elle informe Transformer que les catégories de ce niveau sont identifiées par leurs seules valeurs sources, sans référence à leurs ancêtres. Aucune autre catégorie de ce niveau ne possède la même valeur.

Si une source de données contenant la colonne d'un niveau contient également toutes les colonnes requises par les ancêtres du niveau, il n'est pas nécessaire que le niveau soit unique. Dans ce cas, Transformer peut identifier les catégories du niveau à partir des autres colonnes de la source de données. Néanmoins, quand les colonnes des ancêtres du niveau proviennent d'autres sources de données, les valeurs des catégories de ce niveau doivent être spécifiées comme uniques.

Lorsque vous indiquez que les catégories sont uniques, Transformer ne vérifie pas si c'est le cas. Si vous indiquez qu'un niveau est unique alors qu'il ne l'est pas, Transformer génère des dimensions incorrectes.

Lorsque vous déplacez une catégorie d'un niveau unique, une violation de l'unicité est signalée lors de la génération de catégories, car la catégorie déplacée apparaît alors dans un contexte différent. Utilisez la propriété UniqueMove pour pallier ce problème.

**Paramètre par défaut :** False

## Type

Booléen

## Accès

Lecture/écriture

## Exemples

```
objLocationsDim
= objModel.Dimensions("Sales regions")objLevel
= objLocationsDim.DrillDowns(1).Levels("Employee")objAltDrill
= objLevel.CreateAlternateDrillDownobjNewLevel
= objAltDrill.Levels.Add(xtrObjectType.trLevel)objNewLevel.Unique
= True
```

---

## Propriété UniqueMove

La propriété UniqueMove définit ou retourne une valeur indiquant de quelle manière les niveaux uniques sont traités quand vous déplacez une catégorie.

### Syntaxe

*object* .UniqueMove

### Application

Objet DateLevel

Objet de niveau

### Explication

Utilisez la propriété UniqueMove pour éviter de restructurer manuellement les catégories dans un niveau unique afin de les adapter aux données sources modifiées. Une violation de l'unicité peut survenir lors de la génération de catégorie quand vous déplacez une catégorie d'un niveau unique, car la catégorie déplacée apparaît alors dans un contexte différent.

Affectez la valeur True à cette propriété pour indiquer que ces modifications doivent être traitées comme déplacements uniques. Les valeurs des mesures, même celles ayant été accumulées dans l'ancienne structure de données, sont ensuite ajoutées au nouveau chemin d'accès vers les catégories déplacées.

Cette propriété est disponible uniquement avec les niveaux dont la propriété Unique a pour valeur True.

Paramètre par défaut : False

### Type

Booléen

### Accès

Lecture/écriture

### Exemples

```
objLocationsDim = objModel.Dimensions("Sales regions")
objLevel = objLocationsDim.DrillDowns(1).Levels("Employee") objLevel.UniqueMove
= True
```

---

## Propriété UseAltMDCFile

La propriété UseAltMDCFile définit ou retourne une valeur qui indique si un nom de fichier temporaire peut être utilisé.

### Syntaxe

*object* .UseAltMDCFile

## Application

Objet Cube

Objet CubeGroup

Objet ChildCube

## Explication

Utilisez la propriété UseAltMDCFile pour vérifier s'il est possible d'enregistrer le cube sous un autre nom de fichier.

Si la valeur de cette propriété est définie à True, il est possible d'enregistrer l'objet sous un nom temporaire. Cette propriété est disponible pour les objets Cube, CubeGroup et ChildCube.

Si la valeur retournée par la propriété IsMDCInUse est True, c'est que le cube est en cours d'utilisation par une autre application. Dans ce cas, vous pouvez définir la valeur de la propriété AltMDCFile à True et celle de la propriété AltMDCFile au nom temporaire.

Paramètre par défaut : False

## Type

Booléen

## Accès

Lecture/écriture

---

## Propriété User

La propriété User définit ou retourne un nom d'utilisateur associé à l'espace-noms.

## Syntaxe

*Namespace* .User

## Application

Objet Namespace

## Explication

Cette propriété permet de définir le nom d'un utilisateur afin d'obtenir le CAMID de l'utilisateur depuis la propriété UserCAMID.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
new_namespace = model.Namespaces.Add()new_namespace.User  
= Name_of_User
```

---

## Propriété UserCAMID

La propriété UserCAMID définit ou retourne l'ID utilisateur d'un objet de code d'accès.

## Syntaxe

*Namespace* .UserCAMID

## Application

Objet Namespace

## Explication

Cette propriété permet d'extraire le CAMID d'un utilisateur dans un espace-noms.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

## Exemples

```
new_namespace = model.Namespaces.Add()CAMID_of_User = new_namespace.UserCAMID
```

---

## Propriété UserID

La propriété UserID définit ou retourne l'ID utilisateur d'un code d'accès.

## Syntaxe

*Signon* .UserID

## Application

Objet de code d'accès

## Explication

Transformer utilise les informations des objets de code d'accès pour accéder automatiquement à des bases de données sécurisées. Il est possible que vous ne soyez pas en mesure de changer la valeur de cette propriété si :

- l'objet de code d'accès est un objet de code d'accès IBM Cognos,
- elle référence le code d'accès à un fichier de définition de requête d'Impromptu (.iqd).

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
signon = model.Signons.Add()signon.UserID = "sa"
```

---

## Propriété Value

La propriété Value définit ou retourne la valeur d'invite.

## Syntaxe

*Prompt* .Value

## Application

Objet Prompt

## Explication

Cette propriété est utilisée pour définir ou obtenir la valeur d'invite associée au CurrentValueIndex d'un objet Prompt.

## Type

Chaîne

## Accès

Lecture/écriture

## Exemples

```
new_report = model.Reports.Add()new_query = new_report.Queries.Add()new_prompt.Value = "2005-01-01"
```

---

## Propriété ValuesCount

La propriété ValuesCount renvoie le nombre d'ensembles de valeurs pour l'invite.

## Syntaxe

*Prompt* .ValuesCount

## Application

Objet Prompt

## Explication

Cette propriété peut être utilisée pour effectuer une itération dans les valeurs d'invite conjointement à la propriété CurrentValueIndex.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Entier

## Accès

Lecture

## Exemples

```
new_report = model.Reports.Add()new_query  
= new_report.Queries.Add()new_prompt.CurrentValueIndex  
= indexFor index = 1 To new_prompt.ValuesCount
```

---

## Propriété Version

La propriété Version renvoie le numéro de version de Transformer.

## Syntaxe

*Application* .Version

## Application

Objet Application

## Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Chaîne

## Accès

Lecture

---

## Propriété Views

La propriété Views retourne une collection des objets View associés à une dimension, une dimension de temps ou une vue.

### Syntaxe

*object* .Views

### Application

Objet CustomView

Objet DateDimension

Objet Dimension

### Explication

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Objet

### Accès

Lecture

### Exemples

```
objView = objDimension.Views.Add()
```

---

## Propriété ViewType

La propriété ViewType définit ou renvoie une valeur indiquant si une vue contient toutes, quelques unes ou aucune des catégories d'une dimension.

### Syntaxe

*View* .ViewType

### Application

Objet View

### Explication

Par défaut, la vue d'une dimension est de type trViewTypeAllCategories.

Pour définir une vue personnalisée, affectez la constante trViewTypeCustom à la propriété ViewType. Vous pouvez ensuite utiliser :

- la propriété Apex pour créer une vue d'une catégorie et des catégories enfants,



- la méthode `SetViewStatus` pour définir une vue pour une catégorie ou un niveau.

La propriété `ViewType` utilise les valeurs de la constante `xtrViewType`.

## Type

Constante - `xtrViewType`

## Accès

Lecture/écriture

## Exemples

```
objProductsDim = objModel.Dimensions.Item("Products")objViewItem.ViewType  
= xtrViewType.trViewTypeCustom
```

---

## Propriété `WeekAdd`

La propriété `WeekAdd` définit ou retourne le nombre de jours qui sont ajoutés à une année lunaire.

## Syntaxe

*object* .`WeekAdd`

## Application

Objet `DateDrillDown`

Objet `DateWizard`

## Explication

Lors de l'utilisation de périodes lunaires, une année comprend 52 semaines de 7 jours chacune. Une année lunaire contient donc 364 jours, c'est-à-dire respectivement un ou deux jours de moins qu'une année du calendrier ou une année bissextile.

Utilisez cette propriété pour ajouter une semaine à l'année lunaire. Cela vous permet de synchroniser l'année lunaire sur l'année calendaire. La semaine supplémentaire peut contenir, quatre, cinq, six ou sept jours. Vous pouvez aussi spécifier la valeur `None`.

Transformer ajoute cette semaine supplémentaire au dernier mois ou au dernier trimestre de l'année. De cette manière, chaque année lunaire reste alignée sur le jour de début d'année spécifié, tout en ne s'éloignant pas trop de l'année calendaire.

Si le dernier mois lunaire de l'année possède cinq semaines de type 4-4-5 comme défini par la constante `xtrSpecialFunction`, la semaine supplémentaire du trimestre est ajoutée au mois lunaire qui précède de façon à obtenir une configuration 4-5-5 plutôt qu'une configuration 4-4-6, qui n'est pas valide dans Transformer.

Cette propriété utilise les valeurs de la constante `xtrWeekAdd`.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrWeekAdd

## Accès

Lecture/Ecriture (DateDrillDown)

Ecriture (DateWizard)

## Exemples

```
objDateWizard  
= objModel.DateWizardobjDateDim  
= objDateWizard.CreateDateDimension()objDrillDown  
= objDateDim.DrillDowns(1)objDrillDown.WeekAdd  
= xtrWeekAdd.trWeekAddDays4
```

---

## Propriété WeekSpan

La propriété WeekSpan définit ou retourne une valeur indiquant la manière dont il faut traiter une semaine qui s'étale sur deux années.

## Syntaxe

*object* .WeekSpan

## Application

Objet DateDrillDown

Objet DateWizard

## Explication

Si une dimension de temps contient un niveau de semaine basé sur une année calendaire, la dernière semaine ne se termine généralement pas le même jour que l'année. (Une année lunaire se termine toujours à la fin de la semaine.) La propriété WeekSpan spécifie la manière dont la dernière semaine d'une année doit être décalée ou fractionnée.

Cette propriété utilise les valeurs de la constante xtrWeekSpan.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrWeekSpan

## Accès

Lecture/Ecriture (DateDrillDown)

Ecriture (DateWizard)

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.WeekSpan  
= xtrWeekSpan.trWeekSpanNone
```

---

## Propriété WeekStart

La propriété WeekStart définit ou retourne le premier jour de la semaine.

### Syntaxe

*DateDrillDown* .WeekStart

### Application

Objet DateDrillDown

### Explication

Quand un modèle inclut un niveau hebdomadaire, Transformer utilise cette propriété pour déterminer le premier jour de la semaine.

Cette propriété utilise les valeurs de la constante xtrWeekDay.

Quand vous créez des branches de niveaux secondaires dans une dimension de temps lunaire qui contient des niveaux de détail trimestriels ou inférieurs, les propriétés WeekStart et WeekAdd doit être alignées.

Pour en savoir davantage sur les branches de niveaux secondaires dans les dimensions de temps, reportez-vous à l'aide en ligne de Transformer.

### Type

Constante - xtrWeekDay

## Accès

Lecture/écriture

## Exemples

```
objDateWizard  
= objModel.DateWizardobjDateDim  
= objDateWizard.CreateDateDimension()objDrillDown  
= objDateDim.DrillDowns(1)objDrillDown.WeekStart  
= xtrWeekDay.trMonday
```

---

## Propriété WeekStartDay

La propriété WeekStartDay définit le premier jour de la semaine.

## Syntaxe

*DateWizard* .WeekStartDay

## Application

Objet DateWizard

## Explication

Si vous utilisez l'objet DateWizard pour créer une dimension de temps qui inclut un niveau hebdomadaire, Transformer définit des catégories de semaine sur base de la valeur de cette propriété. Si cette propriété a pour valeur trSunday, par exemple, Transformer spécifie le dimanche comme premier jour de la semaine.

Cette propriété utilise les valeurs de la constante xtrWeekDay.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrWeekDay

## Accès

Ecriture

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.WeekStartDay  
= xtrWeekDay.trMonday
```

---

## Propriété WorkingDay

La propriété WorkingDay définit ou retourne une valeur indiquant si un jour fait partie de la semaine de travail.

## Syntaxe

*object* .WorkingDay(WeekDay)

## Application

Objet DateDimension

Objet DateWizard

## Explication

La propriété WorkingDay permet de déterminer si un jour spécifique fait partie d'une semaine de travail. En comparaison, vous pouvez utiliser la propriété WorkingDays pour définir simultanément plusieurs jours de travail. Vous pouvez aussi définir les propriétés WorkingDays et WorkingDay en tandem pour spécifier une semaine de travail.

Si la dimension de temps inclut un niveau hebdomadaire, tous les jours spécifiés par les propriétés WorkingDay et WorkingDays définissent les jours de la semaine pour lesquels Transformer génère des catégories.

Cette propriété utilise les valeurs de la constante xtrWeekDay.

**Valeur par défaut :** False

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
WeekDay	Requise. Spécifie la valeur de la constante xtrWeekDay. Type : Constante

## Type

Booléen

## Accès

Lecture/Ecriture (DateDimension)

Ecriture (DateWizard)

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.WorkingDays(127)
```

---

## Propriété WorkingDays

La propriété WorkingDays définit ou retourne une valeur indiquant les jours qui font partie de la semaine de travail.

## Syntaxe

```
object .WorkingDays(WeekDays)
```

## Application

Objet DateDimension

Objet DateWizard

## Explication

La propriété WorkingDays permet de spécifier en une seule instruction tous les jours qui constituent une semaine de travail. En comparaison, la propriété WorkingDay permet de déterminer si un jour spécifique fait partie d'une semaine de travail.

Si la dimension de temps inclut un niveau hebdomadaire, tous les jours spécifiés par les propriétés WorkingDays et WorkingDay définissent les jours de la semaine pour lesquels Transformer génère des catégories.

Le paramètre WeekDays peut prendre une des valeurs de la constante xtrWeekday jointes par des signes plus (+), un des nombres ci-dessous ou la somme de deux de ces nombres ou davantage.

- Dimanche = 1
- Lundi = 2
- Mardi = 4
- Mercredi = 8
- Jeudi = 16
- Vendredi = 32
- Samedi = 64

Pour spécifier une semaine de travail du lundi au vendredi, par exemple, entrez 62 (somme de 2, 4, 8, 16 et 32) ou trMonday+trTuesday+trWednesday+trThursday+trFriday.

Pour chacun des sept jours, saisissez 127. Vous pouvez définir les propriétés WorkingDays et WorkingDay en tandem pour spécifier une semaine de travail.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

Paramètre	Description
WeekDays	<p>Requise. Spécifie des valeurs de la constante xtrWeekday jointes par des signes plus (+) ou un nombre correspondant à la somme d'une ou plusieurs valeurs numériques des jours de la semaine.</p> <p><b>Remarque :</b> le mode d'ajout des différentes valeurs de xtrWeekday associées à un signe plus dépend du langage d'implémentation utilisé. Par exemple, dans C#, vous devez procéder comme suit :</p> <pre>(int) xtrWeekDay.trMonday + (int) xtrWeekDay.trWednesday + (int) xtrWeekDay.trFriday;</pre> <p>Type : Long</p>

## Type

Long

## Accès

Lecture/Ecriture (DateDimension)

Ecriture (DateWizard)

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.WorkingDays(127)
```

---

## Propriété YearStartDay

La propriété YearStartDay définit ou retourne le premier jour d'une année.

### Syntaxe

*object* .YearStartDay

### Application

Objet DateDrillDown

Objet DateWizard

### Explication

Utilisez cette propriété pour spécifier la date à laquelle commence une année lorsqu'il ne s'agit pas du 1er janvier comme dans le cas des années fiscales ou lunaires.

Comme les années lunaires contiennent 52 semaines (et non 365 jours), vous devez vous assurer que les propriétés YearStartDay et WeekStart coïncident. Si la propriété YearStartDay a pour valeur 20000104, par exemple, WeekStart doit avoir la valeur trMonday.

La date doit être spécifiée au format AAAAMMJJ.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

### Type

Long

### Accès

Lecture/Ecriture (DateDrillDown)

Ecriture (DateWizard)

### Exemples

```
objDateWizard = objModel.DateWizardobjDateDim  
= objDateWizard.CreateDateDimension()objDrillDown  
= objDateDim.DrillDowns(1)objDrillDown.YearStartDay = 19900101
```

---

## Propriété YearType

La propriété YearType définit le mode de calcul du niveau année d'une dimension de temps.

### Syntaxe

*DateWizard* .YearType

## Application

Objet DateWizard

## Explication

Cette propriété permet de définir une année comme une année calendaire ou une année lunaire.

La propriété YearType utilise les valeurs de la constante xtrSpecialFunction.

Une exception COM est émise en cas d'erreur. Le message transmis avec l'exception varie en fonction de l'erreur.

## Type

Constante - xtrSpecialFunction

## Accès

Ecriture

## Exemples

```
objDateWizard = objModel.DateWizardobjDateWizard.YearType  
= xtrSpecialFunction.trSpecialFunctionYear
```



---

## Chapitre 7. Constantes

Une constante est une valeur fixe qu'il est possible d'utiliser dans une expression.

---

### Liste des valeurs de `xtrAllocationType`

Détermine la manière dont les valeurs sont allouées dans un modèle.

#### Application

Propriété `AllocationType`

Méthode `SetAllocation`

#### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `AllocationType` et la méthode `SetAllocation` des objets `DateDimension`, `Dimension`, `DateLevel`, `Level`, `Category` et `SpecialCategory`.

Si une constante de `xtrAllocationType` n'est pas affectée à un objet `Category` ou `SpecialCategory`, celui-ci hérite du paramètre de l'objet de niveau ou `DateLevel` associé, qui hérite à son tour du paramètre par défaut de l'objet `Dimension` ou `DateDimension` associé.

Si le modèle contient plusieurs sources de données, `Transformer` alloue automatiquement une mesure d'une source de données aux niveaux et aux catégories associés à une autre source de données. Dans ce cas, une valeur de mesure est allouée en tant que constante pour toute la dimension. Cela équivaut à utiliser `trAllocationConstant`. Vous pouvez utiliser `trAllocationNA` pour inverser toute allocation par défaut.

Vous pouvez aussi commander à `Transformer` d'allouer des mesures proportionnellement, en fonction des valeurs d'une autre mesure, en définissant le type `trAllocationByAnotherMeasure` et en identifiant l'autre mesure dans la méthode `SetAllocation`.

Lorsque vous définissez le type d'allocation d'un niveau, il s'applique à partir des catégories de ce niveau jusqu'à toutes les catégories descendantes.

Constante	Description
<code>trAllocationByAnotherMeasure</code>	Base l'allocation sur la valeur d'un autre objet. Vous pouvez allouer proportionnellement la valeur de mesure dans des catégories descendantes en utilisant les valeurs d'une autre mesure.
<code>trAllocationConstant</code>	Utilise la valeur de mesure associée à la catégorie en cours comme valeur de constante pour toutes les catégories descendantes.
<code>trAllocationDefault</code>	Utilise le paramètre d'allocation de l'objet du niveau le plus élevé.

Constante	Description
trAllocationNA	Supprime l'allocation d'une mesure.

## Liste des valeurs de xtrAssociationRole

Détermine le rôle d'un objet Association.

### Application

«Méthode AssociateWith», à la page 133

«Propriété AssociationRole», à la page 210

«Méthode DimensionAssociateWith», à la page 145

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété AssociationRole de l'objet Association, ainsi qu'avec les méthodes AssociateWith et DimensionAssociateWith.

Un objet Association définit la relation entre des objets du modèle et leur source de données sous-jacente. Chaque association possède un rôle. Celui-ci est généralement identique à celui de la source de données, comme indiqué par la constante trAssociationSource. D'autres rôles sont possibles. Celui que l'association prend est limité par la nature des données. Toutes les constantes de xtrAssociationRole ne s'appliquent pas à tous les éléments sources.

Constante	Description
trAssociationCategoryCode	Agit en tant que source pour un code de catégorie unique.
trAssociationCountryCode	Agit en tant que source pour un code de pays ou de région unique.
trAssociationCurrencyDate	Agit en tant que source pour une date de devise.
trAssociationDescription	Agit en tant que source pour une description.
trAssociationDrillThrough	Agit en tant que source pour les informations d'accès au détail.
trAssociationLabel	Agit en tant que source pour un libellé.
trAssociationOrderBy	Agit en tant que source pour les informations de tri.
trAssociationRate	Agit en tant que source pour un taux de conversion.
trAssociationRoleNone (valeur par défaut)	Aucune association n'est définie.
trAssociationShortName	Agit en tant que source pour un nom abrégé.
trAssociationSource	Agit en tant que source pour les données.

---

## Liste des valeurs de `xtrAssociationType`

Détermine le type d'objet Association.

### Application

Propriété `AssociationType`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `AssociationType` de l'objet `Association` pour déterminer le type d'objet `Association`.

Constante	Description
<code>trAssociationQuery</code>	Représente une association avec une source de données.
<code>trAssociationTypeNone</code> (valeur par défaut)	Il n'y a pas d'association.

---

## Liste des valeurs de `xtrCharacterType`

Détermine le jeu de caractères utilisé par un objet.

### Application

Propriété `CharacterType`

Propriété `DataCharacterSet`

### Explication

Utilisez les constantes de cette liste de valeurs avec l'objet `Application` pour spécifier le jeu de caractères que `Transformer` doit employer, ou avec l'objet `FlatFileDataSource` pour désigner le jeu de caractères de la source de données.

Constante	Description
<code>trCharAnsiDoubleByte</code>	Spécifie que l'application Windows d'origine utilise des caractères à octets multiples.
<code>trCharAnsiSingleByte</code>	Spécifie que l'application Windows d'origine n'utilise pas de caractères à octets multiples.
<code>trCharDefault</code>	Spécifie que le paramètre par défaut <code>trCharAnsiDoubleByte</code> est effectif.
<code>trCharOEM</code>	Spécifie que l'application DOS ou OS/2 d'origine utilise le jeu de caractères IBM PC standard.
<code>trUnicode</code>	Spécifie que l'application Windows d'origine utilise des caractères UTF-8 à octets multiples.

---

## Liste des valeurs de xtrCubeConsolidate

Détermine le paramètre de consolidation du cube en cours.

### Application

Propriété Consolidate

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Consolidate des objets Cube, ChildCube et CubeGroup pour déterminer si la consolidation a lieu et de quelle manière. Le processus de consolidation cumule, en un seul enregistrement, les valeurs identiques qui ne sont pas des mesures et récapitule les valeurs de mesure.

Dans le cas d'un groupe de cubes, la propriété Consolidate définit la consolidation par défaut pour chaque cube du groupe. Quand la propriété DuplicateRollup possède une valeur autre que trDuplicateRollupNone, la consolidation a lieu même si la constante trConsolidateNO est effective.

Constante	Description
trConsolidateDefault	Consolide les enregistrements source si cela est jugé utile ou si les données sont triées pour d'autres raisons.
trConsolidateNO	Supprime la consolidation.
trConsolidatePresorted	Consolide les données en double, mais sans trier le fichier source.
trConsolidateYES	Trie le fichier source et consolide les données en double.

---

## Liste des valeurs de xtrCubeCreation

Détermine la manière dont les cubes sont sélectionnés pour le processus de création de cubes.

### Application

Propriété CubeCreation

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété CubeCreation des objets Cube, CubeGroup et ChildCube pour déterminer les cubes à créer avec la méthode CreateMDCFile ou CreateMDCFiles. Les constantes déterminent les PowerCubes du modèle en cours qui sont sélectionnés pour la création de cubes. Pour optimiser la création des cubes, par exemple, utilisez trCubeCreationOFF afin de limiter le processus de création aux seuls cubes qui ont été modifiés.

La liste des valeurs de xtrCubeStatus limite aussi les cubes sélectionnés pour la création.

Constante	Description
trCubeCreationDefault	Utilisez le paramètre du cube parent ou le paramètre ON pour le cube de niveau supérieur.
trCubeCreationOFF	Ne sélectionne pas le cube si les données n'ont pas été modifiées depuis la dernière mise à jour.
trCubeCreationON	Sélectionne le cube. Non valide pour un objet ChildCube.

---

## Liste des valeurs de xtrCubeOptimize

Détermine les paramètres d'optimisation des cubes.

### Application

Propriété Optimize

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Optimize des objets Cube, CubeGroup et ChildCube pour améliorer les performances des PowerCubes et de Transformer. Les constantes représentent des paramètres d'optimisation.

Le paramètre d'optimisation choisi affecte la taille des cubes, la durée de traitement dans Transformer, ainsi que la durée d'accès dans PowerPlay. Sélectionnez le paramètre le plus approprié à votre modèle et aux données.

Constante	Description
trOptimizeAutoPartition	Spécifie que la fonction Auto-partition est disponible. Il s'agit du paramètre par défaut pour les modèles créés dans les versions 6.0 et ultérieures de Transformer.
trOptimizeCategories	Réduit le nombre de catégories dans un cube. Seules les catégories référencées dans la source de données ou spécifiquement désignées pour y être incluses sont ajoutées. Les données de chaque cube sont parcourues une fois de plus pour trouver les catégories requises pour ce cube. Il s'agit du paramètre par défaut pour les modèles créés dans les versions de Transformer antérieures à la 6.0.

Constante	Description
trOptimizeDataPasses	Optimise le nombre de passages à travers les fichiers de travail temporaires lors de la création des cubes. Toutes les catégories sont incluses dans le cube qui est créé, mais celles qui ne sont pas directement référencées ou qui sont référencées indirectement via une catégorie ancêtre ne sont pas visibles dans PowerPlay.
trOptimizeDefault	Définit le paramètre par défaut qui est appliqué à la version de Transformer dans laquelle le modèle a été créé.
trOptimizeDirectCreate	Ajoute au cube toutes les catégories du modèle avant de traiter les sources de données. Il est préférable d'utiliser cette méthode d'optimisation avec des modèles qui génèrent peu de nouvelles catégories et dans lesquels toutes les catégories sont susceptibles d'être ajoutées au cube. Cette fonction n'est pas applicable aux cubes individuels d'un groupe de cubes.

---

## Liste des valeurs de xtrCubeStatus

Détermine le dernier statut de création d'un cube.

### Application

Propriété Status

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Status des objets Cube, CubeGroup et ChildCube pour limiter les cubes à inclure lors de la prochaine exécution de la méthode CreateMDCFile ou CreateMDCFiles. Généralement, ces constantes filtrent les cubes qui ne fonctionnent pas correctement ou sélectionnent ceux qui ont échoué auparavant en vue de les recréer.

La liste des valeurs de xtrCubeCreation filtre aussi les listes de cubes.

Constante	Description
trCubeStatusBUSY	Indique que le cube est en cours de mise à jour.
trCubeStatusFAILED	Indique que le cube n'a pas été correctement mis à jour lors de sa dernière mise à jour.
trCubeStatusINVALID	Indique que le cube n'est pas cohérent avec le modèle.
trCubeStatusMISSING	Indique que le cube existe, mais qu'il n'a pas pu être trouvé ni ouvert.

Constante	Description
trCubeStatusNEW	Indique que le cube est défini dans le modèle, mais qu'il n'a pas été créé.
trCubeStatusOK	Indique que le cube existe et qu'il est exempt d'erreur.
trCubeStatusWARNINGS	Indique que le cube existe, mais que des avertissements ont été détectés lors de sa création ou de sa dernière mise à jour.

---

## Liste des valeurs de `xtrCurrencyTableType`

Détermine si une table des devises s'applique à une devise de base ou à l'euro.

### Application

Propriété `CurrencyTableType`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `CurrencyTableType` de l'objet `CurrencyTable` pour déterminer le type de table des devises.

Constante	Description
trCurrencyTableBase	Désigne une table qui inclut un taux de conversion de base avec lequel d'autres devises sont converties.
trCurrencyTableEuro	Désigne une table qui inclut un taux de conversion de base de l'euro avec lequel d'autres devises sont converties.
trCurrencyTableOther	Indique que la table des devises utilisée est autre que celle de base ou de l'euro.

---

## Liste des valeurs de `xtrDataClass`

Détermine le classement des données d'une colonne.

### Application

Propriété `DataClass`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `DataClass` de l'objet `Column` pour déterminer le type de valeur de données affecté à leur source de données.

Constante	Description
trDataClassArrayMember	Spécifie que la classe de données contient des données issues d'éléments individuels d'un tableau.

Constante	Description
trDataClassDate	Spécifie que la classe de données contient des valeurs de date dans un des formats définis par la liste des valeurs de xtrDateFormat.
trDataClassDefault	La classe de données n'est pas spécifiée dans le modèle. Si une définition du type de données est disponible dans la source, Transformer l'utilise.
trDataClassDescription	Spécifie que la classe de données contient du texte, tel que des libellés, ou des valeurs alphanumériques (telles que des codes).
trDataClassIgnore	Spécifie que le contenu de la source est ignoré et n'est pas traité.
trDataClassQuantity	Spécifie que la classe de données contient des chiffres qui représentent des quantités ou des nombres utilisés comme indicateurs de performances.

---

## Liste des valeurs de xtrDateCategoriesGeneration

Détermine les catégories qui sont générées pour un niveau de date.

### Application

Propriété GenerateCategories

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété GenerateDateCategories de l'objet DateLevel.

Constante	Description
trGenerateDatesAll	Spécifie que toutes les catégories sont générées.
trGenerateDatesDefault	Spécifie que le paramètre par défaut du niveau est utilisé : trGenerateDatesAll pour tous les niveaux, à l'exception du niveau supérieur de chaque branche de niveaux, qui utilise le paramètre par défaut trGenerateDatesNone.
trGenerateDatesNeed	Spécifie que seules les catégories requises par le cube sont générées à partir de la source de données.
trGenerateDatesNone	Spécifie qu'aucune catégorie n'est générée.

---

## Liste des valeurs de xtrDateFormat

Détermine le format d'entrée de date utilisé pour une colonne.

### Application

Propriété DateInputFormat



## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `DateInputFormat` de l'objet `Column` pour déterminer le format de date par défaut appliqué par `Transformer` quand le format de date n'est pas prédéfini dans la source de données. La partie Mois des dates qui possèdent un composant mois peut correspondre à un nombre à deux chiffres ou à une abréviation à trois lettres, selon la source de données.

Constante	Description
<code>trPredefined</code> (valeur par défaut)	Spécifie que le format de date est prédéfini dans la source de données.
<code>trD</code>	Spécifie que la date est au format jour: JJ.
<code>trDMY</code>	Spécifie que la date est au format jour-mois-année: JJMMAAAA
<code>trM</code>	Spécifie que la date est au format mois: MM
<code>trMDY</code>	Spécifie que la date est au format mois-jour-année: MMJJAAAA
<code>trMY</code>	Spécifie que la date est au format mois-année: MMAAAA
<code>trQ</code>	Spécifie que la date est indiquée à l'aide d'un chiffre unique compris entre 1 et 4 pour désigner le trimestre: T
<code>trW</code>	Spécifie que la date est au format semaine.
<code>trY</code>	Spécifie que la date est au format année: AAAA
<code>trYM</code>	Spécifie que la date est au format année-mois: AAAAMM
<code>trYMD</code>	Spécifie que la date est au format année-mois-jour: AAAAMMJJ

---

## Liste des valeurs de `xtrDateLevel`

Détermine le degré de détail des niveaux de date.

### Application

Propriété `DateDegreeofDetail`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `DateDegreeofDetail` de l'objet `Column`.

Les constantes définissent le niveau de date le plus bas (degré de détail) auquel des mesures peuvent être rapportées. Quand un niveau est défini, le rapport des mesures dans la dimension est limité à ce niveau de détail.

Constante	Description
<code>trDateLevelDay</code>	Spécifie que le degré de détail est le jour.
<code>trDateLevelMonth</code>	Spécifie que le degré de détail est le mois.

Constante	Description
trDateLevelQuarter	Spécifie que le degré de détail est le trimestre.
trDateLevelUnspecified	Indique que le degré de détail n'est pas spécifié. Transformer définit le degré de détail en fonction d'autres attributs de la colonne.
trDateLevelWeek	Spécifie que le degré de détail est la semaine.
trDateLevelYear	Spécifie que le degré de détail est l'année.

---

## liste des valeurs de xtrDeployType

Définit le type de déploiement.

### Application

Méthode SetDeployType

### Explication

Utilisez les constantes de cette liste de valeurs avec la méthode SetDeployType de l'objet Cube pour définir le type de déploiement. Ces types correspondent à la stratégie de déploiement du PowerCube. Si vous utilisez trDeployType\_SWAPSINGLE, l'échec d'un déploiement à un emplacement donné n'affecte pas le déploiement aux autres emplacements. Cette action correspond à la commande de l'interface utilisateur **Copier dans les emplacements disponibles, puis activer**. Si vous utilisez trDeployType\_SWAPTOGETHER, tout échec de déploiement à un emplacement engendre l'annulation du déploiement aux autres emplacements, même si ceux-ci avaient abouti. Cette action correspond à la commande de l'interface utilisateur **Copier dans tous les emplacements, puis activer**.

Constante	Description
trDeployType_DEFAULT	Spécifie le type de déploiement par défaut.
trDeployType_NONE	Spécifie aucun déploiement.
trDeployType_NULL	Spécifie le type de déploiement null.
trDeployType_SWAPSINGLE	Indique de copier dans les emplacements disponibles, puis d'activer.
trDeployType_SWAPTOGETHER	Indique de copier dans tous les emplacements, puis d'activer.

---

## Liste des valeurs de xtrDuplicateRollup

Détermine la manière dont Transformer récapitule les enregistrements dupliqués qui sont retournés par la source de données.

## Application

Propriété DuplicateRollup

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété DuplicateRollup de l'objet Measure pour définir les options de cumul.

Constante	Description
trDuplicateRollupAverage	Spécifie que le cumul utilise la moyenne de toutes les valeurs de la mesure trouvée dans les enregistrements en double.  Si la propriété RegularRollup a pour valeur trRollupAverage, le cumul des doubles est effectué en premier lieu.
trDuplicateRollupFirst	Spécifie que le cumul utilise la première valeur de mesure trouvée dans les enregistrements en double, en suivant l'ordre des enregistrements dans la source de données.
trDuplicateRollupLast	Spécifie que le cumul utilise la dernière valeur de mesure trouvée dans les enregistrements en double, en suivant l'ordre des enregistrements dans la source de données.
trDuplicateRollupMaximum	Spécifie que le cumul utilise la valeur de mesure la plus élevée trouvée dans les enregistrements en double.
trDuplicateRollupMinimum	Spécifie que le cumul utilise la plus petite valeur de mesure trouvée dans les enregistrements en double.
trDuplicateRollupNone	Spécifie que le cumul des doubles n'est pas effectué. Cette option utilise la constante de xtrRollup spécifiée pour la propriété RegularRollup.
trDuplicateRollupSum	Spécifie que le cumul cumule les valeurs de mesure trouvées dans les enregistrements en double.

---

## Liste des valeurs de xtrGenerateOptions

Détermine l'option de génération des catégories dans une dimension de temps.

## Application

Propriété GenerateTimePeriod

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété GenerateTimePeriod de l'objet DateDimension pour déterminer les catégories à générer dans la dimension.

Constante	Description
trGenerateAll	Spécifie que toutes les catégories sont générées.
trGenerateDefault	Spécifie que le paramètre par défaut du niveau est utilisé: trGenerateAll pour tous les niveaux, à l'exception du niveau supérieur de chaque branche de niveaux, qui utilise le paramètre par défaut trGenerateNone.
trGenerateNeed	Spécifie que seules les catégories requises par le cube sont générées à partir de la source de données.
trGenerateNone	Spécifie qu'aucune catégorie n'est générée.

## Liste des valeurs de xtrInclusion

Détermine si une catégorie est incluse dans un modèle ou un cube.

### Application

Propriété DrillInclusion

Propriété Inclusion

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété DrillInclusion des objets DrillDown et DateDrillDown, ainsi qu'avec la propriété Inclusion des objets Category, SpecialCategory, DateLevel et de niveau. Les constantes déterminent dans quelles circonstances des catégories sont incluses dans un modèle ou un cube.

Si une constante de xtrInclusion n'est pas affectée à un objet Category ou SpecialCategory, la catégorie hérite du paramètre de l'objet de niveau ou DateLevel associé.

Quand un cube doit être complété, Transformer inclut des ancêtres d'une catégorie, quel que soit le paramètre d'inclusion utilisé.

Constante	Description
trInclusionDefault	Niveau : trInclusionWhenNeeded. Catégorie : utilise le paramètre du niveau dans lequel se trouve la catégorie.
trInclusionExclude	Exclut la catégorie, ses descendants, ainsi que toutes les données associées provenant du cube.  Afin d'utiliser cette consante : la catégorie ne peut pas être la catégorie de référence d'une autre catégorie ou ne peut pas être liée à une catégorie spéciale.
trInclusionGenerate	Inclut toujours la catégorie.

Constante	Description
trInclusionSuppress	Niveau : conserve les catégories dans le modèle, mais exclut des cubes celles dont la valeur d'origine est en blanc.  Catégories : exclut la catégorie du cube, mais inclut tous les descendants et leurs valeurs.  Afin d'utiliser cette consante : la catégorie ne peut pas être la catégorie de référence d'une autre catégorie ou ne peut pas être liée à une catégorie spéciale.
trInclusionWhenNeeded	Inclut la catégorie dans le modèle, mais pas dans le cube lorsqu'elle possède la valeur zéro.

---

## Liste des valeurs de xtrMeasureType

Détermine le type et l'usage d'un objet Measure.

### Application

Propriété MeasureType

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété MeasureType de l'objet Measure pour déterminer le type de mesure qui a été créé. Les objets Measure peuvent avoir les types suivants : ordinaire, calculé et nombre de catégories.

Constante	Description
trCalculatedMeasure	Spécifie que la mesure dérive des valeurs de la propriété ExpressionText.
trCountMeasure	Spécifie que la mesure dérive des valeurs de la propriété CategoryCountLevel.
trRegularMeasure (valeur par défaut)	Spécifie que la mesure est une mesure ordinaire qui dérive des valeurs d'une association.

---

## Liste des valeurs de xtrMissingValue

Détermine la manière dont PowerPlay indique les valeurs manquantes.

### Application

Propriété MissingValue

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété MissingValue de l'objet Measure pour déterminer ce qui apparaît dans PowerPlay quand la valeur numérique d'une mesure est absente de la source de données.

Constante	Description
trMissingValueNA	Spécifie que l'abréviation 'NA' est utilisée. De cette façon, les éléments manquants ne sont pas mal interprétés comme étant des valeurs nulles.
trMissingValueZERO (valeur par défaut)	Spécifie que des zéros sont utilisés pour les valeurs manquantes.

---

## Liste des valeurs de xtrObjectType

Détermine le type d'objet.

### Application

Méthode Add (Objects)

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Type des objets de Transformer et le paramètre Type des méthodes Add et Item. Vous pouvez, par exemple, utiliser les constantes de xtrObjectType avec la méthode Add pour ajouter de nouveaux objets à une collection.

Constante	Description
trApplication	Spécifie que l'objet est l'application Transformer.
trAssociation	Indique qu'il s'agit d'un objet Association.
trCalculationDefinition	Indique qu'il s'agit d'un objet CalculationDefinition.
trCategory	Indique qu'il s'agit d'un objet Category.
trCategorySet	Indique qu'il s'agit d'un objet CategorySet.
trChildCube	Indique qu'il s'agit d'un objet ChildCube.
trColumn	Indique qu'il s'agit d'un objet Column.
trCrossTabDataSource	Indique qu'il s'agit d'un objet CrossTabDataSource.
trCube	Indique qu'il s'agit d'un objet Cube.
trCubeGroup	Indique qu'il s'agit d'un objet CubeGroup.
trCurrencyRate	Indique qu'il s'agit d'un objet CurrencyRate.
trCurrencyRecord	Indique qu'il s'agit d'un objet CurrencyRecord.
trCurrencyTable	Indique qu'il s'agit d'un objet CurrencyTable.
trCustomView	Indique qu'il s'agit d'un objet CustomView.

Constante	Description
trDateDimension	Indique qu'il s'agit d'un objet DateDimension.
trDateDrillDown	Indique qu'il s'agit d'un objet DateDrillDown.
trDateLevel	Indique qu'il s'agit d'un objet DateLevel.
trDateWizard	Indique qu'il s'agit d'un objet DateWizard.
trDbDataSource	Indique qu'il s'agit d'un objet DbDataSource.
trDimension	Indique qu'il s'agit d'un objet Dimension.
trDrillDown	Indique qu'il s'agit d'un objet DrillDown.
trDrillThroughTarget	Indique qu'il s'agit d'un objet DrillThroughTarget.
trFlatFileDataSource	Indique qu'il s'agit d'un objet FlatFileDataSource.
trFilter	Indique qu'il s'agit d'un objet Filter.
trIqdDataSource	Indique qu'il s'agit d'un objet IqdDataSource.
trLevel	Indique qu'il s'agit d'un objet de niveau.
trMeasure	Indique qu'il s'agit d'un objet Measure.
trModel	Indique qu'il s'agit d'un objet Model.
trName	Indique qu'il s'agit d'un objet Name.
trNamespace	Indique qu'il s'agit d'un objet Namespace.
trNoType	Spécifie que le type d'objet est non identifié.
trPackage	Indique qu'il s'agit d'un objet Package.
trPrompt	Indique qu'il s'agit d'un objet Prompt.
trReport	Indique qu'il s'agit d'un objet Report.
trSecurityObject	Indique qu'il s'agit d'un objet SecurityObject.
trSignon	Indique qu'il s'agit d'un objet de code d'accès.
trSpecialCategory	Indique qu'il s'agit d'un objet SpecialCategory.
trSuspendedModel	Indique qu'il s'agit d'un objet SuspendedModel.
trView	Indique qu'il s'agit d'un objet View.

---

## Liste des valeurs de xtrOrigin

Détermine l'origine d'une colonne.

### Application

Propriété Origin

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Origin de l'objet Column pour déterminer l'origine de la colonne, notamment si elle est dérivée d'une source de données ou d'un calcul.

Constante	Description
trOriginCalculated	Spécifie que la colonne est basée sur un calcul.
trOriginGenerated	Spécifie que la colonne est issue des catégories générées.
trOriginManual	Spécifie que la colonne a été créée manuellement dans Transformer.
trOriginSource	Spécifie que la colonne est issue d'une source de données IQD.
trOriginUnspecified	Spécifie que l'origine est inconnue.

---

## Liste des valeurs de xtrPowerCubeGeneration

Détermine à quel moment une source de données est référencée par un modèle.

### Application

Propriété GeneratePowerCube

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété GeneratePowerCube des objets CrossTabDataSource, DataSource, DbDataSource, FlatFileDataSource, IqdDataSource et Query.

Les constantes de xtrPowerCubeGeneration permettent de déterminer si une source de données est référencée pendant la génération d'une catégorie, la création d'un cube, les deux, ou si elle n'est pas référencée du tout. Utilisez les constantes de cette liste de valeurs dans des modèles qui possèdent plusieurs sources de données pour éviter tout traitement superflu ainsi que pour optimiser l'efficacité du modèle.

Constante	Description
trGenerationCreatePowerCubes	<p>Spécifie que Transformer lit la source de données pour générer des catégories et/ou créer des cubes comme requis, même pour les sources de données purement structurales.</p> <p>Utilisez cette option dans un environnement de conception et de développement ou pour créer des cubes comportant des mesures basées sur le nombre d'enregistrements plutôt que sur les valeurs de mesure des enregistrements eux-mêmes.</p>



Constante	Description
trGenerationDefault	Spécifie que Transformer lit toutes les colonnes de la source de données liées aux niveaux du modèle pour déterminer s'ils sont associées à des mesures. Si tel est le cas, le cube est créé ou mis à jour à l'aide des valeurs de mesure. Si la source de données est purement structurelle, Transformer ne génère que des catégories.  Utilisez cette option dans un environnement de production ainsi que pour les sources de données transactionnelles et structurelles contenant des données non statiques.
trGenerationGenerateCategories	Spécifie que Transformer ne lit la source de données que pour les informations structurelles ainsi que pour générer des catégories.
trGenerationNoCreatePowerCubes	Spécifie que Transformer n'accède pas à la source de données quand il crée des catégories et des cubes.

---

## Liste des valeurs de xtrPreferences

Détermine le format de date, la journalisation des erreurs et les préférences de tri d'une application.

### Application

Propriété DateFormat

Propriété LogErrorLevel

Propriété SortComparisonRule

### Explication

Utilisez les constantes de cette liste de valeurs avec les propriétés DateFormat, LogErrorLevel et SortComparisonRule de l'objet Application.

Toutes les constantes de cette liste de valeurs ne s'appliquent pas aux trois propriétés.

Constante	Description
trDateFormatFromControlPanel	Spécifie que le format de date est défini dans les paramètres régionaux du Panneau de configuration de Windows. Cette option s'applique à la propriété DateFormat.

Constante	Description
trDateFormatPredefined	Spécifie que le format de date est défini dans la source de données utilisée par le modèle. Cette option s'applique à la propriété DefaultDateFormat.
trLogErrorsAndAbove	Indique si tous les messages d'erreur grave et les messages d'erreur doivent être consignés dans le fichier journal. Les erreurs se produisent au niveau de la transaction et entraînent le marquage des cubes comme erronés. Cette option s'applique à la propriété LogErrorLevel.
trLogInformationAndAbove	Spécifie si tous les messages doivent être consignés dans le fichier journal. Cette option s'applique à la propriété LogErrorLevel.
trLogSevereErrors	Spécifie si seuls les messages d'erreur grave doivent être consignés dans le fichier journal. Les erreurs graves peuvent être provoquées par une restriction d'administration système de l'unité centrale, du disque, d'un fichier ou des ressources de transaction ou par l'altération d'un modèle, d'un cube ou d'un fichier temporaire. Cette option s'applique à la propriété LogErrorLevel.
trLogWarningsAndAbove	Spécifie si les messages d'erreur grave, les messages d'erreur et les messages d'avertissement doivent être consignés dans le fichier journal. Les avertissements n'ont aucune incidence sur le traitement, mais font ressortir les problèmes potentiels. Cette option s'applique à la propriété LogErrorLevel.
trSortFromControlPanel	Spécifie que l'application utilise la règle de comparaison associée aux paramètres régionaux du Panneau de configuration de Windows. Cette option s'applique à la propriété SortComparisonRule.
trSortIgnoreControlPanel	Spécifie que l'application utilise la règle de comparaison interne de Transformer, à savoir la comparaison de chaque octet de la représentation binaire de chaque chaîne. Cette option s'applique à la propriété SortComparisonRule.

---

## liste des valeurs de xtrPromptValueType

Détermine le type d'invite.

## Application

Propriété PromptValueType

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété PromptValueType de l'objet Prompt pour définir le type d'invite.

Constante	Description
trSingleValuePrompt	Spécifie qu'il s'agit d'une invite de choix d'une valeur.
trMultiValuePrompt	Spécifie qu'il s'agit d'une invite de choix de plusieurs valeurs.
trBoundedRangePrompt	Spécifie qu'il s'agit d'une invite liée.
trUnboundedRangePrompt	Spécifie qu'il s'agit d'une invite non liée.

---

## Liste des valeurs de xtrRollup

Détermine la manière dont Transformer cumule les valeurs de mesure de niveaux inférieurs à des niveaux de catégories supérieurs.

## Application

Propriété RegularRollup

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété RegularRollup de l'objet Measure pour définir les options de cumul.

Constante	Description
trRollupAny	Détermine s'il existe des valeurs de catégorie.  Dans PowerPlay, cette option affiche 1 si des enregistrements d'une catégorie descendante contiennent des valeurs ou 0 s'il n'existe pas d'enregistrement pour cette mesure, ou bien si tous les enregistrements existants possèdent des valeurs nulles pour cette mesure.
trRollupAverage	Génère une moyenne des valeurs de tous les enregistrements des catégories descendantes de la catégorie active.
trRollupCount	Génère le nombre d'enregistrements qui contiennent des valeurs non nulles dans toutes les catégories descendantes de la catégorie active.

Constante	Description
trRollupCount All	Génère le nombre d'enregistrements, y compris ceux qui contiennent des valeurs nulles pour cette mesure, pour toutes les catégories descendantes de la catégorie active.
trRollupDefault	Spécifie que l'option trRollupSum est utilisée.
trRollupExternal	Génère des valeurs sources qui ont été directement affectées à des enregistrements de données spécifiques.
trRollupMaximum	Génère la valeur maximale des données parmi toutes les catégories descendantes de la catégorie active.
trRollupMinimum	Génère la valeur minimale des données parmi toutes les catégories descendantes de la catégorie active.
trRollupSum	Génère la somme des valeurs de toutes les catégories descendantes de la catégorie active.

## Liste des valeurs de xtrRollupTiming

Détermine à quel moment les calculs sont effectués sur les mesures calculées.

### Application

Propriété RollupTiming

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété RollupTiming de l'objet Measure. Les constantes déterminent à quel moment ont lieu les cumuls des mesures par rapport aux calculs impliquant ces dernières. La propriété MeasureType de l'objet Measure doit avoir la valeur trCalculatedMeasure.

Constante	Description
trTimingAfterRollup	Spécifie que les mesures calculées sont basées sur les valeurs cumulées. La séquence est la suivante : <ol style="list-style-type: none"> <li>1 Le cumul ordinaire s'effectue sur toutes les mesures.</li> <li>2 Le cumul d'état temporel est effectué.</li> <li>3 Les mesures allouées sont calculées.</li> <li>4 Les calculs sont réalisés sur les mesures calculées.</li> </ol>

Constante	Description
trTimingBeforeRollup	Spécifie que les mesures calculées sont basées sur les valeurs sources d'origine. La séquence est la suivante :  1 Les calculs sont réalisés lors de la création du cube.  2 Le cumul ordinaire s'effectue sur toutes les mesures.  3 Le cumul d'état temporel est effectué.  4 Les mesures allouées sont calculées.
trTimingDefault	Spécifie que l'option trTimingAfterRollup est utilisée.

---

## Liste des valeurs xtrSecurityType

Détermine le type d'objet de sécurité.

### Application

Propriété Type

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété Type de l'objet SecurityObject pour définir le type d'objet de sécurité.

Constante	Description
trSecurityType_User	Indique que l'objet de sécurité est un utilisateur.
trSecurityType_Group	Indique que l'objet de sécurité est un groupe.
trSecurityType_Role	Indique que l'objet de sécurité est un rôle.

---

## Liste des valeurs de xtrSourceType

Détermine le type de fichier représenté par une source de données.

### Application

Propriété SourceType

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété SourceType des objets CrossTabDataSource, DataSource, DbDataSource, FlatFileDataSource et IqdDataSource pour déterminer la source externe des données et des définitions de données.

Constante	Description
trAccess	Spécifie que la source est un fichier de base de données Access.
trAccessQuery	Spécifie que la table de source de données est décrite dans un fichier de requête Access (.mdb). Les valeurs d'entrée sont extraites d'une base de données reconnue par ODBC via l'exécution d'une requête SQL stockée dans le fichier de requête Access.
trClipper	Spécifie que la source est un fichier Clipper.
trDBase	Spécifie que la source est un fichier de base de données dBase.
trDictionary	Spécifie que la source est un dictionnaire de données.
trExcelCrossTab	Indique que la source est un fichier de tableau de feuilles de calcul Excel.
trExcelDatabase	Spécifie que la source est un fichier de base de données de feuille de calcul Excel.
trFixedAscii	Spécifie que la source est un fichier texte comportant un enregistrement par ligne. Chaque zone occupe un nombre d'octets donné. Un séparateur de ligne de texte signale la fin de chaque enregistrement.
trFixedAsciiNoCRLF	Spécifie que la source est un fichier texte comportant un enregistrement par ligne. Chaque zone occupe un nombre d'octets donné. La fin de chaque enregistrement n'est pas signalée par un séparateur de ligne de texte.
trFlatFile	Spécifie que la source est un fichier texte comportant un enregistrement par ligne. Les zones sont délimitées par le caractère spécifié dans la propriété FieldSeparator.
trFlatFileColumnNames	Spécifie que la source est un fichier texte comportant un enregistrement par ligne et où la première ligne représente des noms de colonne. Les zones (valeurs de colonne) sont délimitées par le caractère spécifié dans la propriété FieldSeparator.
trFoxPro	Spécifie que la source est un fichier de base de données FoxPro.
trLotus123CrossTab	Indique que la source est un fichier de tableau de feuilles de calcul Lotus 1-2-3.

Constante	Description
trLotus123Database	Indique que la source est un fichier de base de données de feuilles de calcul Lotus 1-2-3.
trParadox	Spécifie que la source est un fichier de base de données Paradox.
trPowerHousePortable	Indique que la source est un sous-fichier transférable de PowerHouse.
trQuery	Indique que la source est un fichier de définition de requête d'Impromptu (.iqd). Les valeurs d'entrée sont extraites d'une base de données Impromptu prise en charge en exécutant une requête SQL définie dans Impromptu.  Le contenu du fichier de définition de requête d'Impromptu est stocké dans le modèle et les contenus incorporés sont actualisés chaque fois que vous générez des catégories ou que vous créez des cubes.

---

## Liste des valeurs de xtrSpecialFunction

Détermine le type de calendrier utilisé avec les dates.

### Application

Propriété DateFunction

Propriété MonthType

Propriété QuarterType

Propriété YearType

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété DateFunction de l'objet DateLevel, ainsi qu'avec les propriétés MonthType, QuarterType et YearType de l'objet DateWizard. Les constantes déterminent si un niveau de date utilise un calendrier standard ou lunaire ainsi que le format de la date.

Puisque les segments du calendrier lunaire sont plus courts que ceux du calendrier standard, des jours non affectés subsistent en fin d'année. Vous pouvez ajuster ceux-ci à l'aide de la propriété WeekAdd et des constantes de la liste des valeurs de xtrWeekAdd.

Constante	Description
trSpecialFunctionDay	Spécifie qu'il s'agit d'un jour standard au format AAAA-MMM-JJ ou AA-MMM-JJ.

Constante	Description
trSpecialFunctionLunarMonth	Spécifie que le mois est basé sur un mois lunaire, qui comprend exactement 4 semaines.
trSpecialFunctionLunarMonth445	Spécifie que le mois est basé sur la répétition de 3 séquences de mois lunaire contenant respectivement 4 semaines, 4 semaines et 5 semaines.
trSpecialFunctionLunarMonth454	Spécifie que le mois est basé sur la répétition de 3 séquences de mois lunaire contenant respectivement 4 semaines, 5 semaines et 4 semaines.
trSpecialFunctionLunarMonth544	Spécifie que le mois est basé sur la répétition de 3 séquences de mois lunaire contenant respectivement 5 semaines, 4 semaines et 4 semaines.
trSpecialFunctionLunarQuarter	Spécifie que le trimestre est basé sur des trimestres lunaires, qui comprennent exactement 13 semaines.
trSpecialFunctionLunarYear	Spécifie que l'année est basée sur un calendrier lunaire, qui comprend exactement 52 semaines.
trSpecialFunctionMonth	Spécifie que le mois est basé sur le calendrier standard, au format AAAA-MMM ou AA-MMM.
trSpecialFunctionNone	Spécifie que la date est basée sur la colonne source du niveau.
trSpecialFunctionNull	Spécifie qu'aucun paramètre de date n'est défini.
trSpecialFunctionQuarter	Spécifie que le trimestre est basé sur le calendrier standard sous la forme AAAA T ou AA T, où T représente le numéro du trimestre (1, 2, 3 ou 4).
trSpecialFunctionWeek	Spécifie qu'il s'agit d'une semaine du calendrier standard au format AAAA-MMM-JJ ou AA-MMM-JJ.
trSpecialFunctionYear	Spécifie que l'année est basée sur le calendrier standard au format AAAA ou AA.

---

## Liste des valeurs de xtrStorage

Détermine la manière dont les branches de niveaux numériques sont triées dans les niveaux ainsi que la manière dont les données de mesure sont stockées.

### Application

Propriété OrderByStorageType

Propriété StorageType



## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `StorageType` de l'objet `Measure` pour déterminer comment Transformer stocke des valeurs de mesure dans des fichiers de travail servant à créer des cubes.

Vous pouvez aussi utiliser une de ces constantes avec la propriété `OrderByStorageType` pour définir un tri de catégories quand celui-ci est basé sur une colonne avec une classe de données de type numérique. Par exemple, vous basez un tri de produits sur la quantité vendue et la mesure appelée 'Quantité', qui est associée à la colonne `Quantité`, possède un stockage de type "entier 16 bits".

Constante	Description
<code>trStorageBigFloat</code>	Stocke les nombres avec une plage de 1,7E +/- 308 sous forme d'entier 64 bits à virgule flottante.
<code>trStorageBigInt</code>	Stocke les nombres compris entre -2 147 483 648 et +2 147 483 647 sous forme d'entier 32 bits.
<code>trStorageDefault</code>	Fournit un stockage par défaut en fonction du type de données. Pour les données numériques, le stockage par défaut est <code>trStorageBigInt</code> .
<code>trStorageSmallInt</code>	Stocke les nombres compris entre -32 768 et +32 767 sous forme d'entier 16 bits.

---

## Liste des valeurs de `xtrTimeAggregate`

Détermine le type de période de date relative.

### Application

Propriété `Aggregate`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `Aggregate` de l'objet `SpecialCategory` pour personnaliser des catégories spéciales de date relative. Ces catégories effectuent le suivi des mesures pour des périodes spécifiques par rapport à la date en cours. Ces constantes spécifient le type de période de date relative sur lequel vous basez la catégorie spéciale.

Un objet `SpecialCategory` de date relative personnalisée requiert aussi la définition des paramètres d'autres propriétés, telles que `ContextOffset` et `TargetOffset`, qui fournissent la plage de dates et le contexte dans lesquels la propriété `Aggregate` fonctionne.

Constante	Description
<code>trAggregateNone</code>	Spécifie qu'aucune période de date cumulée n'est définie.
<code>trAggregateRunning</code>	Représente un nombre de périodes de date se terminant à une période N spécifique, indiquée par des propriétés telles que <code>ContextOffset</code> et <code>TargetOffset</code> .

Constante	Description
trAggregateRunningGrouped	Produit le même résultat que trAggregateRunning, mais en créant une plage de catégories de période N.
trAggregateSingle	Représente une période de date unique définie par des propriétés telles que ContextOffset et TargetOffset.
trAggregateToDate	Représente un ensemble séquentiel de périodes commençant au début d'une période et se terminant à une autre période spécifique. Les périodes de date sont définies par des propriétés telles que ContextOffset et TargetOffset.
trAggregateToDateGrouped	Produit le même résultat que trAggregateToDate, mais en créant une plage de catégories de date de fin.

---

## Liste des valeurs de xtrTimeArrayType

Détermine les paramètres des tableaux de dates.

### Application

Propriété TimeArrayType

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété TimeArrayType de l'objet Column pour déterminer si un tableau de dates contient des valeurs trimestrielles ou mensuelles.

Constante	Description
trTimeArrayMonth	Spécifie qu'il s'agit de la première des douze colonnes adjacentes qui composent un tableau mensuel.
trTimeArrayNA	Spécifie que l'objet n'est pas associé à un tableau de dates.
trTimeArrayQuarter	Spécifie qu'il s'agit de la première des quatre colonnes adjacentes qui composent un tableau trimestriel.

---

## Liste des valeurs de xtrTimeRollup

Détermine la manière dont Transformer cumule les mesures à des périodes de temps spécifiques.

### Application

Propriété TimeStateRollup

## Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `TimeStateRollup` de l'objet `Measure` pour définir les options de cumul des valeurs de date.

Constante	Description
<code>trTimeRollupAverage</code>	Utilise la moyenne des valeurs de mesure parmi toutes les catégories de la période examinée.
<code>trTimeRollupCurrentPeriod</code>	<p>Utilise la valeur de mesure issue de la catégorie désignée comme "période en cours" dans la dimension de temps. Si la période en question ne comprend pas la période en cours, le résultat est identique à celui de la dernière période.</p> <p>Par exemple, une dimension de temps contient les années, les trimestres et les mois, le trimestre 1 commençant en janvier. La période en cours est définie sur Avril 2000.</p> <p>Au niveau de l'année, PowerPlay affiche la valeur de mesure pour avril 2000. Au niveau du trimestre, il affiche la valeur de mesure pour avril dans le trimestre 2 (car avril est la période en cours), mais également la valeur du dernier mois actif dans tous les autres trimestres (c'est-à-dire, mars pour le trimestre 1, septembre pour le trimestre 3 et décembre pour le trimestre 4).</p>
<code>trTimeRollupEarlyPeriod</code>	<p>Utilise la valeur de mesure issue de la première période subordonnée dans la période examinée.</p> <p>Par exemple, si une dimension de temps comprend les années, les trimestres et les mois et si vous examinez les données au niveau du trimestre, PowerPlay affiche, pour chaque trimestre, la valeur de mesure à partir du premier mois de chaque trimestre.</p> <p>Quand vous étudiez les données au niveau Année, PowerPlay affiche la première valeur du premier mois pour le premier trimestre de chaque année.</p>

Constante	Description
trTimeRollupLatePeriod	Utilise la valeur de mesure issue de la dernière période subordonnée dans la période examinée.  Par exemple, si une dimension de temps comprend les années, les trimestres et les mois et si vous examinez les données au niveau du trimestre, PowerPlay affiche, pour chaque trimestre, la valeur de mesure à partir du dernier mois de chaque trimestre.  Quand vous étudiez les données au niveau Année, PowerPlay affiche la valeur du dernier mois pour le dernier trimestre de chaque année.
trTimeRollupMaximum	Utilise la plus grande valeur de mesure parmi toutes les catégories de la période examinée.
trTimeRollupMinimum	Utilise la plus petite valeur de mesure parmi toutes les catégories de la période examinée.
trTimeRollupNone	Spécifie que le cumul d'état temporel n'est pas effectué. Cette option utilise la valeur définie pour la propriété RegularRollup.

## Liste des valeurs de xtrTimeType

Détermine le paramètre de degré de détail des dates.

### Application

Propriété EnableTimePeriod

Propriété GenerateTimePeriod

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété EnableTimePeriod de l'objet DateWizard et la propriété GenerateTimePeriod de l'objet DateDimension. Ces constantes indiquent le niveau auquel Transformer autorise le rapport à partir des colonnes relatives à la date dans une source de données. Par exemple, une colonne de date contenant des valeurs pour l'année, le trimestre et le mois peut être limitée de sorte à n'autoriser que des rapports trimestriels. Vous ne pouvez pas spécifier une période de rapport inférieure au plus bas niveau défini par la propriété DateInputFormat.

Constante	Description
trTimeTypeDay	Spécifie que la période de rapport minimale est le jour.
trTimeTypeMonth	Spécifie que la période de rapport minimale est le mois.

Constante	Description
trTimeTypeQuarter	Spécifie que la période de rapport minimale est le trimestre.
trTimeTypeWeek	Spécifie que la période de rapport minimale est la semaine.
trTimeTypeYear	Spécifie que la période de rapport minimale est l'année.

## Liste des valeurs de `xtrViewStatus`

Détermine les catégories à inclure dans une vue.

### Application

Méthode `GetViewStatus`

Méthode `SetViewStatus`

### Explication

Utilisez les constantes de cette liste de valeurs avec la méthode `SetViewStatus` des objets `Category` et `SpecialCategory`, ainsi qu'avec la méthode `GetViewStatus` de l'objet `View`. Ces constantes décrivent les catégories qui sont incluses quand un `PowerCube` est créé à partir de la vue.

Constante	Description
trViewStatusApexAncestor	Ne contient que la catégorie du sommet et ses descendants directs.
trViewStatusClear	Spécifie qu'aucune restriction n'est définie.
trViewStatusCloakMom	Spécifie que la catégorie et tous les descendants sont omis, mais que leurs valeurs sont conservées pour être cumulées dans des catégories de niveau supérieur.
trViewStatusClipMom	Spécifie que la catégorie et tous ses descendants sont exclus.
trViewStatusInvisibleKid	Indique que la catégorie est exclue d'un ancêtre <code>Cloaked</code> ou <code>Summary</code> .
trViewStatusRemoveKid	Indique que la catégorie est exclue d'un ancêtre <code>Excluded</code> .
trViewStatusSummaryMom	Inclut une catégorie avec un résumé des données de tous ses descendants.
trViewStatusSuppressed	Spécifie que la catégorie n'est pas incluse, mais que les catégories parents et enfants le sont.

---

## Liste des valeurs de `xtrViewType`

Détermine le type de vue.

### Application

Propriété `DimensionInclude`

Propriété `DimensionViewType`

Propriété `ViewType`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `DimensionViewType` des objets `Cube` et `CubeGroup`, ainsi qu'avec la propriété `ViewType` de l'objet `View` et enfin avec la propriété `DimensionInclude`.

Quand vous créez une dimension, le cube associé contient une vue des catégories importantes dans cette dimension. Cette vue peut contenir toutes les catégories, certaines d'entre elles ou aucune d'elles.

Constante	Description
<code>trViewTypeAllCategories</code>	Spécifie que la vue contient la dimension et toutes les catégories.
<code>trViewTypeCustom</code>	Spécifie que la vue ne contient que les catégories sélectionnées.
<code>trViewTypeNone</code>	Spécifie que l'objet n'est pas associé à une vue.
<code>trViewTypeOmitDimension</code>	Spécifie que la totalité de la dimension est ignorée.

---

## Liste des valeurs de `xtrWeekAdd`

Détermine le nombre de jours à ajouter à une année lunaire pour égaler une année calendaire.

### Application

Propriété `WeekAdd`

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété `WeekAdd` de l'objet `DateWizard` ou `DateDrillDown` pour synchroniser les années lunaires avec les années calendaires.

Une année lunaire contient 364 jours, c'est-à-dire un ou deux jours de moins qu'une année du calendrier ou une année bissextile, respectivement. Dans `Transformer`, vous pouvez ajouter une semaine supplémentaire de 4 à 7 jours à l'année lunaire. L'application ajoute alors cette semaine à la fin du dernier mois ou du dernier trimestre de l'année.

Constante	Description
trWeekAddNone (valeur par défaut.)	Spécifie que des semaines complètes ou partielles ne sont jamais ajoutées.
trWeekAddDays7	Ajoute 7 jours.
trWeekAddDays6	Ajoute 6 jours.
trWeekAddDays5	Ajoute 5 jours.
trWeekAddDays4	Ajoute 4 jours.

---

## Liste des valeurs de xtrWeekDay

Détermine si un jour fait partie de la semaine de travail.

### Application

Propriété WeekStart

Propriété WorkingDay

Propriété WorkingDays

### Explication

Utilisez les constantes de cette liste de valeurs avec les propriétés WorkingDay et WorkingDays de l'objet DateWizard ou DateDimension, ainsi qu'avec la propriété WeekStart de l'objet DateDrillDown.

Dans le cas d'une dimension de temps, ces constantes indiquent si un jour fait partie d'une semaine de travail et s'avèrent utiles pour créer des semaines de travail non standard. Les constantes spécifient aussi les jours de la semaine pour lesquels Transformer génère des catégories dans la dimension.

Dans le cas d'une branche de niveaux de dates, ces constantes déterminent le jour qui marque le premier jour des catégories de semaine dans les périodes de haut niveau.

Constante	Description
trSunday	Spécifie que dimanche fait partie de la semaine de travail.
trMonday	Spécifie que lundi fait partie de la semaine de travail.
trTuesday	Spécifie que mardi fait partie de la semaine de travail.
trWednesday	Spécifie que mercredi fait partie de la semaine de travail.
trThursday	Spécifie que jeudi fait partie de la semaine de travail.
trFriday	Spécifie que vendredi fait partie de la semaine de travail.
trSaturday	Spécifie que samedi fait partie de la semaine de travail.

---

## Liste des valeurs de xtrWeekspan

Détermine la manière dont il faut définir une semaine qui s'étale sur deux années.

### Application

Propriété WeekSpan

### Explication

Utilisez les constantes de cette liste de valeurs avec la propriété WeekSpan de l'objet DateWizard ou DateDrillDown pour déterminer où il faut ajuster la dernière semaine de l'année.

Si une dimension de temps contient un niveau de semaine dans une année calendaire, la dernière semaine ne se termine généralement pas le même jour que l'année. (Une année lunaire se termine toujours à la fin de la semaine.) Transformer peut décaler ou répartir la dimension en fonction du paramètre de la propriété WeekSpan.

Constante	Description
trWeekSpanFirstPeriod	Décale la semaine dans l'année où elle débute.
trWeekSpanLargerPeriod	Décale la semaine dans l'année comportant le plus grand nombre de jours de cette semaine.
trWeekSpanLastPeriod	Décale la semaine dans l'année où elle se termine.
trWeekSpanNone (valeur par défaut.)	Spécifie qu'une semaine qui est à cheval sur deux années est ignorée.
trWeekSpanSplitAll	Répartit la semaine en deux semaines partielles distinctes et applique une partie à chaque année sur laquelle s'étend la semaine.
trWeekSpanSplitMost	Répartit la semaine à cheval sur les deux années, à condition que chaque partie contienne au moins deux jours ; sinon la semaine est placée dans l'année comportant le plus grand nombre de jours de cette semaine.



---

## Chapitre 8. Equivalents IU

L'automatisation OLE d'IBM Cognos Transformer utilise une interface en langage de programmation comme alternative à l'interface utilisateur de Transformer. Le présent document comprend des informations relatives aux équivalents IU de la plupart des méthodes et propriétés OLE qui vous permettent de vous familiariser avec Transformer. Toutefois, gardez à l'esprit que les équivalents IU des éléments ajoutés dans les éditions 7.3 et ultérieures ne sont pas répertoriés.

Elle présente un modèle Transformer sous la forme d'un ensemble de collections et d'objets qui sont modifiés par des propriétés et sur lesquels agissent des méthodes. Utilisez-la pour créer et gérer des dimensions, des niveaux, des sources de données, des mesures, des catégories, des branches de niveaux, ainsi que tous les autres objets du modèle, voire pour créer des PowerCubes.

Quand vous créez un modèle dans OLE, vous devez créer vos objets et leur affecter des valeurs dans un ordre hiérarchique. Par exemple, avant de pouvoir créer un niveau, vous devez créer la dimension dans laquelle il figure. Pour en savoir davantage sur les hiérarchies d'objets, reportez-vous à Grille de hiérarchie d'objets Transformer.

---

### Collections

La table suivante présente un récapitulatif des équivalences de l'interface utilisateur pour les collections.

Collections	Equivalence de l'interface utilisateur
Collection Associations	Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)  Feuille des propriétés du niveau des données, onglet Source  Feuille des propriétés de la mesure, onglet Type (quand l'option Type a pour valeur Colonne)  Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée)
Collection CalculationDefinitions	Feuille des propriétés de la dimension, onglet "Calcul"
Collection Categories	Fenêtre des dimensions, fenêtre des catégories

<b>Collections</b>	<b>Equivalence de l'interface utilisateur</b>
Collection CategorySets	Boîte de dialogue Calcul d'une catégorie de dimension (accessible par l'intermédiaire de la feuille des propriétés de la dimension, onglet Calcul)
Collection ChildCubes	Liste des PowerCubes
Collection Columns	Liste des sources de données
Collection Cubes	Liste des PowerCubes
Collection CurrencyRates	Feuille des propriétés de la table des devises
Collection CurrencyRecords	Feuille des propriétés de la table des devises
Collection CurrencyTables	Feuille des propriétés de la table des devises
Collection DataSources	Liste des sources de données
Collection DimensionLevels	Grille des dimensions
Collection Dimensions	Grille des dimensions
Collection DrillDowns	Fenêtre des dimensions, fenêtre des catégories
Collection LevelDrillDowns	Grille des dimensions
Collection Levels	Grille des dimensions
Collection Measures	Liste des mesures
Collection Names	N'importe quelle fenêtre de Transformer contenant des objets
Collection Signons	Menu <b>Affichage</b> , commande <b>Codes d'accès</b> , liste des codes d'accès
Collection SuspendedModels	Menu <b>Fichier</b> , commande <b>Afficher les modèles en attente</b> , zone <b>Sélection de modèles en attente</b>
Collection Views	Fenêtre des dimensions, onglet <b>Dimension</b>

## Objets

La table suivante présente un récapitulatif des équivalences de l'interface utilisateur pour les objets.

Objets	Equivalent IU
Application	Fenêtre IBM Cognos Transformer
Association	<p>Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)</p> <p>Feuille des propriétés du niveau des données, onglet Source</p> <p>Feuille des propriétés de la table des devises (quand la case à cocher <b>Utiliser une source de données des devises externe</b> est activée)</p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b>)</p>
CalculationDefinition	<p>Feuille des propriétés de la dimension, onglet "Calcul"</p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b>, bouton <b>Calcul</b>, boîte de dialogue <b>Calcul d'une mesure</b></p>
Catégorie	Fenêtre des catégories
CategorySet	Feuille des propriétés de la dimension, onglet <b>Calcul</b> , bouton <b>Ajouter</b> ou <b>Modifier</b> , bouton <b>Calcul</b> , boîte de dialogue <b>Calcul d'une catégorie de dimension</b> , sous-fenêtre <b>Composants disponibles</b>
ChildCube	Liste des PowerCubes (groupe de cubes)
Colonne	Liste des sources de données (quand l'objet source de données est développé)
CrossTabDataSource	Liste des sources de données
Cube	Liste des PowerCubes
CubeGroup	Liste des PowerCubes
CurrencyRate	Feuille des propriétés de la table des devises
CurrencyRecord	Feuille des propriétés de la table des devises

Objets	Equivalent IU
CurrencyTable	Feuille des propriétés de la table des devises
CustomView	Liste Vue personnalisée
DataSource	Liste des sources de données
DateDimension	Grille des dimensions
DateDrillDown	Fenêtre des dimensions
DateLevel	Grille des dimensions
DateWizard	Menu <b>Outils</b> , commande <b>Assistant Dates</b>
DbDataSource	Liste des sources de données
Dimension	Grille des dimensions
DrillDown	Fenêtre des dimensions
Filtres	Liste des sources de données
FlatFileDataSource	Liste des sources de données
IqdDataSource	Liste des sources de données
Niveau	Grille des dimensions
Mesure	Liste des mesures
Modèle	Menu ?, commande A propos de <nom de fichier modèle>
Nom	<Objet> feuille de propriété, zone Nom
Pack	Liste des sources de données
Demandes	Liste des sources de données
Query	Liste des sources de données
Rapport	Liste des sources de données
DrillThroughTarget	Cible d'accès au détail
SecurityObject	Fenêtre Gestion des objets de sécurité, ouverte à partir de l'option Afficher les objets de sécurité du menu Sécurité.
Signon	Menu <b>Affichage</b> , commande <b>Codes d'accès</b> , liste des codes d'accès

Objets	Equivalent IU
SpecialCategory	Fenêtre des dimensions
SuspendedModel	Menu <b>Fichier</b> , commande <b>Afficher les modèles en attente</b> , zone <b>Sélection de modèles en attente</b>
Vue	Fenêtre des dimensions, onglet <b>Dimension</b>

## Méthodes

La table suivante présente un récapitulatif des équivalences de l'interface utilisateur pour les méthodes.

Méthodes	Equivalent IU
Ajout	Menu Edition, commande Insérer <i>objet</i>
AssociateWith	<p>Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)</p> <p>Feuille des propriétés du niveau, onglet <b>Source</b> ou onglet <b>Classement</b></p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b>)</p> <p>Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée)</p>
CheckLocalPowerCubes	Menu <b>Outils</b> , commande <b>Vérifier les PowerCubes locaux</b>
CheckModel	Menu <b>Outils</b> , commande <b>Vérifier le modèle</b>
CleanHouse	Menu <b>Outils</b> , commande <b>Supprimer les catégories</b>
Fermer	Menu <b>Fichier</b> , commande <b>Fermer</b>
ConnectWithCategory	Fenêtre des dimensions (lors du déplacement d'une catégorie vers une nouvelle catégorie parent du même niveau)

Méthodes	Equivalent IU
CreateAlternateDrillDown	Menu <b>Edition</b> , commande <b>Créer une branche de niveaux</b> (quand un niveau est sélectionné)
CreateDateDimension	Assistant Dates, bouton <b>Terminer</b>
CreateMDCFile	Menu <b>Exécuter</b> , commande <b>Créer le PowerCube sélectionné</b>
CreateMDCFiles	Menu <b>Exécuter</b> , commande <b>Créer les PowerCubes</b>
Supprimer	Menu <b>Edition</b> , commande <b>Supprimer &lt; objet &gt;</b>
DimensionAssociateWith	Assistant Dates
DoAutoDesign	Assistant Nouveau modèle, case à cocher <b>Conception automatique</b>
GenerateCategories	Menu <b>Exécuter</b> , commande <b>Générer les catégories</b>
Elément	Une catégorie, une source de données, une dimension ou un cube sélectionné
LoadCurrencyTable	Feuille des propriétés de la table des devises, bouton <b>Charger la table</b> (quand la case à cocher <b>Utiliser une source de données des devises externe</b> est activée)
Connexion	Menu <b>Sécurité</b> , option <b>Connexion</b>
Déplacer	Toute utilisation qualifiée de la méthode glisser-déplacer
MoveToCategory	Fenêtre des dimensions (pendant le glissement d'une catégorie vers une nouvelle position dans la hiérarchie)
MoveToLevel	Fenêtre des dimensions (pendant le glissement d'une catégorie vers une nouvelle position dans la hiérarchie)
NewModel	Menu <b>Fichier</b> , commande <b>Nouveau</b>
OpenModel	Menu <b>Fichier</b> , commande <b>Ouvrir</b>
Supprimer	Menu <b>Edition</b> , commande <b>Supprimer &lt; objet &gt;</b>
ResetPartitions	Menu <b>Outils</b> , commande <b>Supprimer les partitions</b>

Méthodes	Equivalent IU
Enregistrer	Menu <b>Fichier</b> , commande <b>Enregistrer</b>
SaveAs	Menu <b>Fichier</b> , commande <b>Enregistrer sous</b>
SetAllocation	Feuille des propriétés de l'objet, onglet <b>Allocation</b> , bouton <b>OK</b> (lorsqu'il y a plusieurs sources de données)
TestBuild	Menu <b>Exécuter</b> , commande <b>Jeu d'essai</b>
Mettre à jour	Feuille des propriétés de l'objet, bouton <b>OK</b>

## Propriétés

La table suivante présente un récapitulatif des équivalences de l'interface utilisateur pour les propriétés.

Propriétés	Equivalent IU
ActivityMeasure	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Mesure d'activité</b> Accès : Lecture/Ecriture
Aggregate	Feuille des propriétés de la catégorie spéciale, onglet Date relative, zone Type Accès : Lecture/Ecriture
AllocationMeasure	Feuille des propriétés de la dimension, du niveau ou de la catégorie, onglet <b>Allocation</b> Accès : Lecture
AllocationType	Feuille des propriétés de la dimension, du niveau ou de la catégorie, onglet <b>Allocation</b> Accès : Lecture
AllowCurrencyConversion	Feuille des propriétés de la mesure, onglet <b>Général</b> , case à cocher <b>Permettre la conversion monétaire</b> Accès : Lecture/Ecriture
AllowDrillThrough	Feuille des propriétés de l'objet, onglet <b>Accès au détail</b> , case à cocher <b>Autoriser l'accès au détail pour cette mesure</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
AlternateQueryPath	Feuille des propriétés du PowerCube, onglet <b>Général</b> , zone <b>Fichier source</b> Accès : Lecture/Ecriture
Apex	Menu <b>Diagramme</b> , commande <b>Sommet</b> (quand une catégorie d'une vue est sélectionnée) Accès : Lecture/Ecriture
Application	Interface utilisateur de Transformer Accès : Lecture
AssociationRole	Feuille des propriétés de la dimension, onglet <b>Général</b> , zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données) Feuille des propriétés du niveau des données, onglet <b>Source</b> Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b> ) Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée) Accès : Lecture/Ecriture
Associations	Feuille des propriétés de la dimension, onglet <b>Général</b> , zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données) Feuille des propriétés du niveau des données, onglet <b>Source</b> Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b> ) Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée) Accès : Lecture



Propriétés	Equivalent IU
AssociationType	<p>Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)</p> <p>Feuille des propriétés du niveau des données, onglet Source</p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b>)</p> <p>Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée)</p> <p>Accès : Lecture/Ecriture</p>
BlankSubstitute	<p>Feuille des propriétés du niveau, onglet <b>Général</b>, zone <b>Remplacement des blancs</b></p> <p>Accès : Lecture/Ecriture</p>
BlockParentTotals	<p>Feuille des propriétés du PowerCube, onglet <b>Traitement</b>, case à cocher <b>Exclure les totaux des parents dont les enfants sont exclus</b></p> <p>Accès : Lecture/Ecriture</p>
CacheCrossTabs	<p>Feuille des propriétés du PowerCube, onglet <b>Traitement</b>, case à cocher <b>Permettre la mise en cache des tableaux</b></p> <p>Accès : Lecture/Ecriture</p>
CalculationDefinitions	<p>Feuille des propriétés de la dimension, onglet "Calcul"</p> <p>Accès : Lecture</p>
CanAllocate	<p>Feuille des propriétés de la dimension, du niveau ou de la catégorie (quand l'onglet <b>Allocation</b> est visible)</p> <p>Accès : Lecture</p>
CanAllocateByMeasure	<p>Feuille des propriétés de l'objet, onglet <b>Allocation</b>, case à cocher <b>Sélection d'une mesure</b></p> <p>Accès : Lecture</p>

Propriétés	Equivalent IU
CanAllocateMeasure	Feuille des propriétés de la dimension, du niveau ou de la catégorie, onglet <b>Allocation</b> Accès : Lecture
Categories	Fenêtre des dimensions, fenêtre des catégories Accès : Lecture
Categorie	Fenêtre des dimensions, fenêtre des catégories Accès : Lecture/Ecriture
CategoryCount	Menu <b>Edition</b> , commande <b>Afficher les nombres</b> Accès : Lecture
CategoryCountLevel	Feuille des propriétés de la mesure, onglet <b>Type</b> , zone <b>Dimension</b> et zone <b>Niveau</b> (quand le bouton <b>Nombre de catégories</b> est sélectionné) Accès : Lecture/Ecriture
CategorySets	Boîte de dialogue Calcul d'une catégorie de dimension (accessible par l'intermédiaire de la feuille des propriétés de la dimension, onglet Calcul) Accès : Lecture
CharacterType	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Jeu de caractères</b> Accès : Lecture/Ecriture
ChildCategories	Fenêtre des dimensions, fenêtre des catégories Accès : Lecture
ChildCubes	Liste des PowerCubes Accès : Lecture
ChildCustomViews	Liste Vues personnalisées et fenêtre des catégories, onglet Vue personnalisée
Code	Feuille des propriétés de la catégorie, onglet <b>Général</b> , zone <b>Code de catégorie</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
Columns	Liste des sources de données Accès : Lecture
ColumnsLoaded	Liste des sources de données Accès : Lecture
Consolidate	Feuille des propriétés du PowerCube, onglet <b>Général</b> , zone <b>Consolidation</b> Accès : Lecture/Ecriture
Context	Feuille des propriétés du niveau, onglet <b>Classement</b> , liste des branches de niveaux Accès : Lecture/Ecriture
ContextLevel	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Période de référence</b> Accès : Lecture/Ecriture
ContextOffset	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Décalage</b> Accès : Lecture/Ecriture
ConvergenceLevel	Dans la grille des dimensions, le niveau auquel se rencontrent deux ou plusieurs branches de niveaux secondaires Accès : Lecture
Count	Aucun Accès : Lecture
CountryCode	Feuille des propriétés de la devise, rôle du code de pays ou boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Code du pays</b> Accès : Lecture/Ecriture
CubeCreation	Feuille des propriétés du PowerCube, onglet Traitement, zone Création de cube: Activée ou Désactivée Accès : Lecture/Ecriture
Cubes	Liste des PowerCubes Accès : Lecture

Propriétés	Equivalent IU
CubeStamp	Aucun
CurrencyDecimals	Feuille des propriétés de la devise, boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Positions décimales</b> Accès : Lecture/Ecriture
CurrencyFormatOverride	Feuille des propriétés de la devise, boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Supplanter le code du pays</b> Accès : Lecture/Ecriture
CurrencyIsEMU	Feuille des propriétés de la devise, boîte de dialogue <b>Enregistrement de devise</b> , case à cocher <b>Appartient à l'Union économique et monétaire (UEM)</b> Accès : Lecture/Ecriture
CurrencyIsEuro	Aucun Accès : Lecture/Ecriture
CurrencyRates	Feuille des propriétés de la table des devises Accès : Lecture
CurrencyRecord	Feuille des propriétés de la devise, boîte de dialogue <b>Enregistrement de devise</b> Accès : Lecture
CurrencyRecords	Feuille des propriétés de la table des devises Accès : Lecture
CurrencySymbol	Feuille des propriétés de la devise, boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Symbole de la devise</b> Accès : Lecture/Ecriture
CurrencyTable	Menu <b>Fichier</b> , commande <b>Table des devises</b> , feuille des propriétés de la table des devises Accès : Lecture
CurrencyTables	Menu <b>Fichier</b> , commande <b>Table des devises</b> , feuille des propriétés de la table des devises Accès : Lecture

Propriétés	Equivalent IU
CurrencyTableType	Feuille des propriétés de la table des devises, table de base ou table de l'euro Accès : Lecture/Ecriture
CurrentModel	Menu ?, commande A propos de <nom de fichier modèle> Nom de fichier Name Accès : Lecture
CustomViews	Liste Vues personnalisées et fenêtre des catégories, onglet Vue personnalisée
DataCharacterSet	Feuille des propriétés des préférences, onglet <b>Général</b> , zone <b>Jeu de caractères des données</b> Accès : Lecture/Ecriture
DataClass	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Classe de données</b> Accès : Lecture/Ecriture
DataRange	Assistant Nouvelle source de données ou Assistant Nouveau modèle, zone <b>Table ou plage</b> Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Table ou plage</b> Accès : Lecture/Ecriture
DataSourcePath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Source de données</b> Accès : Lecture/Ecriture
DataSources	Liste des sources de données Accès : Lecture
DataTemporaryFilesPath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Fichiers temporaires de données (rép1;rép2 )</b> Accès : Lecture/Ecriture
DateDegreeofDetail	Feuille des propriétés de la colonne, onglet <b>Date</b> , zone <b>Degré de détail</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
DateDegreeofDetailLevelName	Feuille des propriétés de la colonne, onglet <b>Général</b> , zone <b>Degré de détail</b> (quand la case à cocher <b>Contient des valeurs de mesure avec cumul calculé à l'externe</b> est activée dans l'onglet <b>Général</b> de la feuille des propriétés de la source de données)  Accès : Lecture/Ecriture
DateFormat	Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Format de date</b>  Accès : Lecture/Ecriture
DateFunction	Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Fonction de date</b>  Accès : Lecture/Ecriture
DateInputFormat	Feuille des propriétés de la colonne, onglet <b>Date</b> , zone <b>Format de date</b>  Accès : Lecture/Ecriture
DateLevel	Feuille des propriétés de la table des devises, boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Niveau de date</b>  Accès : Lecture/Ecriture
DateWizard	Menu <b>Outils</b> , commande <b>Assistant Dates</b>  Accès : Lecture
DecimalPoint	Feuille des propriétés de la source de données (type de source de données de fichier à plat), onglet <b>Général</b> , zone <b>Séparateur décimal</b>  Accès : Lecture/Ecriture
Decimals	Objet Column : Aucun  Accès : Lecture
DefaultDateFormat	Feuille des propriétés des préférences, onglet <b>Général</b> , zone <b>Format par défaut des dates</b>  Accès : Lecture/Ecriture
Description	Feuille des propriétés de l'objet, onglet <b>Description</b>  Accès : Lecture/Ecriture

Propriétés	Equivalent IU
DesiredPartitionSize	Feuille des propriétés du PowerCube, onglet <b>Auto-partition</b> , zone <b>Taille souhaitée des partitions</b> Accès : Lecture/Ecriture
DetachDataSource	Feuille des propriétés des préférences, onglet <b>Fichiers</b> , case à cocher <b>Libérer la source de données après génération des catégories</b> Accès : Lecture/Ecriture
DetailLevel	Feuille des propriétés du PowerCube (groupe de cubes), onglet <b>Groupe</b> , zone <b>Degré de détail des catégories du niveau</b> Accès : Lecture/Ecriture
DimensionInclude	Digramme de dimension, sous-fenêtre gauche, onglet <b>Classe d'utilisateurs</b> Accès : Lecture/Ecriture
DimensionLevels	Grille de dimensions Accès : Lecture
DimensionName	Assistant Dates Accès : écriture
Dimensions	Grille de dimensions Accès : Lecture
DimensionView	Fenêtre des dimensions, onglet <b>Classe d'utilisateurs</b> Accès : Lecture
DimensionViewType	Feuille des propriétés du PowerCube, onglet <b>Dimensions</b> (quand vous cliquez avec le bouton droit de la souris sur une dimension) Accès : Lecture/Ecriture
DrillCode	Feuille des propriétés de la catégorie de branche, zone <b>Code de catégorie</b> Accès : Lecture/Ecriture
DrillDowns	Grille de dimensions Accès : Lecture

Propriétés	Equivalent IU
DrillInclusion	Feuille des propriétés de la catégorie de branche, onglet <b>Général</b> , zone <b>Inclusion</b> Accès : Lecture/Ecriture
DrillThroughTargets	Feuille des propriétés du PowerCube, onglet <b>Accès au détail</b> , zone <b>Rapports personnalisés</b> Accès : Lecture
DuplicateRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Cumul des doubles</b> Accès : Lecture/Ecriture
DuplicateWeight	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Pondération des doubles</b> Accès : Lecture/Ecriture
EarliestDate	Assistant Dates Feuille des propriétés de la dimension (Date), onglet <b>Date</b> , zone <b>Date la plus ancienne</b> Menu <b>Exécuter</b> , commande <b>Générer les catégories de date</b> , zone <b>Date de début</b> Access : écriture (Assistant Dates) lecture/écriture (DateDimension)
EMUEntryDate	Feuille des propriétés de la table des devises, boîte de dialogue <b>Enregistrement de devise</b> , zone <b>Date d'entrée dans l'UEM (AAAAMMJJ)</b> Accès : Lecture/Ecriture
EnableMessageLogging	Feuille des propriétés des préférences, onglet <b>Journalisation</b> , case à cocher <b>Activer la journalisation de messages</b> Accès : Lecture/Ecriture
EnableTimePeriod	Assistant Dates Accès : écriture
EstimatedRows	Feuille des propriétés du PowerCube, onglet <b>Auto-partition</b> , zone <b>Nombre approximatif d'enregistrements consolidés</b> Accès : Lecture/Ecriture



Propriétés	Equivalent IU
ExcludeAutoPartition	<p>Feuille des propriétés de la dimension, onglet <b>Général</b>, case à cocher <b>Exclure la dimension de la partition automatique</b></p> <p>Accès : Lecture/Ecriture</p>
ExpressionText	<p>Feuille des propriétés de la colonne, onglet <b>Général</b>, option <b>Calculé</b></p> <p>Feuille des propriétés de la dimension, onglet <b>Calcul</b>, bouton <b>Ajouter</b>, bouton <b>Calcul</b>, boîte de dialogue <b>Calcul d'une catégorie de dimension</b></p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b>, option <b>Calculée</b>, bouton <b>Calcul</b>, boîte de dialogue <b>Calcul d'une mesure</b></p> <p>Accès : Lecture/Ecriture</p>
External	<p>Feuille des propriétés de la source de données, onglet <b>Général</b>, case à cocher <b>Contient des valeurs de mesure avec cumul calculé à l'externe</b></p> <p>Accès : Lecture/Ecriture</p>
FieldSeparator	<p>Feuille des propriétés de la source de données (type de source de données de fichier à plat), onglet <b>Source</b>, zone <b>Séparateur de zones</b></p> <p>Accès : Lecture/Ecriture</p>
FileName	<p>Explorateur Windows</p> <p>Accès : Lecture/Ecriture</p>
Format	<p>Feuille des propriétés de l'objet, onglet <b>Format</b></p> <p>Accès : Lecture/Ecriture</p>
FormatDecimals	<p>Feuille des propriétés de la mesure, onglet <b>Format</b>, zone <b>Positions décimales</b></p> <p>Accès : Lecture/Ecriture</p>
FullName	<p>Explorateur Windows</p> <p>Accès : Lecture</p>
GenerateCategories	<p>Feuille des propriétés de la source de données, onglet <b>Général</b>, case à cocher <b>Générer les catégories uniquement</b></p> <p>Accès : Lecture/Ecriture</p>

Propriétés	Equivalent IU
GenerateDateCategories	Feuille des propriétés du niveau (Date), onglet <b>Date</b> , case à cocher <b>Générer toutes les catégories de la période</b> (activé en cas de sélection de trGenerateDatesAll)  Accès : Lecture/Ecriture
GeneratePowerCube	Feuille des propriétés de la source de données, onglet <b>Général</b> , case à cocher <b>Au moment de la création du PowerCube</b>  Accès : Lecture/Ecriture
GenerateTimePeriod	Menu <b>Exécuter</b> , <b>Générer les catégories de date</b> (quand une dimension de temps est sélectionnée)  Accès : Lecture/Ecriture
Group	Zone <b>Définition du calcul</b> , case à cocher <b>Regrouper les catégories calculées</b>  Accès : Lecture/Ecriture
GroupLevel	Feuille des propriétés du PowerCube, onglet <b>Groupe</b> , zone <b>Niveau</b>  Accès : Lecture/Ecriture
HasSubdimension	Dans la grille des dimensions, le nom d'un niveau qui contient une sous-dimension est suivi de trois points de suspension (...)  Accès : Lecture
Inclusion	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Inclusion</b>  Accès : Lecture/Ecriture
IncrementalUpdate	Feuille des propriétés du PowerCube, onglet <b>Traitement</b> , case à cocher <b>Mise à jour incrémentielle de ce cube</b>  Accès : Lecture/Ecriture
InputScale	Feuille des propriétés de la colonne, onglet <b>Général</b> , zone <b>Facteur d'échelle en entrée</b>  Accès : Lecture/Ecriture
IsAnyColumnMismatched	Menu <b>Outils</b> , commande <b>Vérifier les colonnes</b>  Accès : Lecture

Propriétés	Equivalent IU
IsBad	Menu <b>Fichier</b> , commande <b>Afficher les modèles en attente</b> Accès : Lecture
IsExpressionValid	Bouton <b>OK</b> de Feuille des propriétés de la dimension, onglet <b>Calcul</b> , bouton <b>Ajouter</b> ou <b>Modifier</b> , boîte de dialogue <b>Calcul d'une catégorie de dimension</b>  Feuille des propriétés de la colonne, onglet <b>Général</b> , bouton <b>Calcul</b> , boîte de dialogue <b>Calcul d'une colonne</b> (quand la colonne est de type <b>Calculée</b> ),  Feuille des propriétés de la mesure, onglet <b>Type</b> , bouton <b>Calcul</b> , boîte de dialogue <b>Calcul d'une mesure</b> (quand la mesure est de type <b>calculé</b> ). Accès : Lecture
IsManual	Dans la grille des dimensions, les niveaux manuels sont représentés par une icône en forme de main Accès : Lecture
IsolationLevel	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Niveau d'isolement</b> (quand la source de données est une IQD)  Assistant Nouveau modèle Accès : Lecture/Ecriture
IsPrimary	Feuille des propriétés de l'objet, onglet <b>Général</b> , case à cocher <b>Catégorie de branche principale</b> Accès : lecture/écriture (DateDrillDown et DrillDown) lecture (Category)
KeyName	Feuille des propriétés de la catégorie, onglet <b>Général</b> , zone <b>Valeur d'origine</b> Accès : Lecture/Ecriture
Label	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Libellé de l'&lt;objet&gt;</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
LastUseDate	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Dernière utilisation</b> Accès : Lecture
LatestDate	Assistant Dates Feuille des propriétés de la dimension de date, onglet <b>Date</b> , zone <b>Date la plus récente</b> Menu <b>Exécuter</b> , commande <b>Générer les catégories de date</b> , zone <b>Date de fin</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDimension)
LevelDrillDowns	Grille des dimensions, vue des branches de niveaux secondaires Accès : Lecture
Levels	Grille des dimensions, vue des niveaux d'une dimension Accès : Lecture
LocalPath	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Fichier de données local</b> Accès : Lecture/Ecriture
LogErrorLevel	Feuille des propriétés des préférences, onglet "Journalisation", une des quatre options : Messages d'erreur grave, Messages d'erreur et précédents, Messages d'avertissement et précédents ou Messages d'information et précédents Accès : Lecture/Ecriture
LogFileAppend	Feuille des propriétés des préférences, onglet <b>Journalisation</b> , case à cocher <b>Ajouter l'information à la fin du fichier journal</b> Accès : Lecture/Ecriture
LogFileName	Feuille des propriétés des préférences, onglet <b>Journalisation</b> , zone <b>Nom du fichier journal</b> Accès : Lecture/Ecriture
LogFilesPath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Fichiers journaux</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
Lunar	Aucun (utiliser la feuille des propriétés du niveau, onglet <b>Date</b> , zone <b>Fonction de date</b> pour déterminer la valeur) Accès : Lecture/Ecriture
ManualCurrentPeriod	Feuille des propriétés de la dimension, onglet <b>Date</b> , case à cocher <b>Définir automatiquement la période en cours</b> Accès : Lecture/Ecriture
MaximizeSpeed	Feuille des propriétés de la source de données, onglet <b>Général</b> , zone <b>Vérification de l'unicité</b> Accès : Lecture/Ecriture
MaxNumPartLevels	Feuille des propriétés du cube, onglet <b>Auto-partition</b> , zone <b>Nombre maximal d'étapes</b> Accès : Lecture/Ecriture
MaxTransactionNumber	Feuille des propriétés des préférences, onglet <b>Général</b> , zone <b>Nombre maximal de transactions par enregistrement</b> Accès : Lecture/Ecriture
MDCFile	Feuille des propriétés du PowerCube, onglet <b>Sortie</b> , zone <b>Nom de fichier du PowerCube</b> Accès : Lecture/Ecriture
MeasureInclude	Feuille des propriétés du PowerCube, onglet <b>Mesures</b> Accès : Lecture/Ecriture
MeasureName	Feuille des propriétés du PowerCube, onglet <b>Général</b> , zone <b>Nom de la mesure</b> Accès : Lecture/Ecriture
Measures	Liste des mesures Accès : Lecture
MeasureType	Feuille des propriétés de la mesure, onglet <b>Type</b> , <b>Colonne</b> , <b>Calculée</b> ou <b>Nombre de catégories</b> Accès : Lecture

Propriétés	Equivalent IU
MissingValue	Feuille des propriétés de la mesure, onglet <b>Général</b> , zone <b>Valeurs manquantes</b> Accès : Lecture/Ecriture
ModelName	Menu <b>Fichier</b> , commande <b>Afficher les modèles en attente</b> , zone <b>Sélection de modèles en attente</b> Accès : Lecture/Ecriture
ModelsPath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Modèles</b> Accès : Lecture/Ecriture
ModelTemporaryFilePath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Fichiers temporaires de modèles</b> Accès : Lecture/Ecriture
ModelType	Explorateur Windows Accès : Lecture
MonthType	Assistant Dates Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Fonction de date</b> Accès : écriture
Name	Feuille des propriétés de l'objet, <objet> zone Nom Feuille des propriétés de l'objet, <objet> zone Libellé Accès : lecture/écriture lecture (Application, Nom, Rapport)
NewCatsLocked	Feuille des propriétés de l'objet, onglet <b>Général</b> , case à cocher <b>Empêcher la création automatique de nouvelles catégories</b> Accès : Lecture/Ecriture
Optimize	Feuille des propriétés du PowerCube, onglet <b>Traitement</b> , zone <b>Optimisation</b> Accès : Lecture/Ecriture
OrderByDescending	Feuille des propriétés du niveau, onglet <b>Classement</b> , zone <b>Ordre du classement</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
OrderByStorageType	Feuille des propriétés du niveau, onglet <b>Classement</b> , zone <b>Type de classement</b> Accès : Lecture/Ecriture
Origin	Aucun Accès : Lecture
OriginalName	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Nom original</b> Accès : Lecture
Orphanage	Feuille des propriétés de la catégorie, case à cocher <b>Orphelinat</b> Accès : Lecture/Ecriture
OutputScale	Feuille des propriétés de la mesure, onglet <b>Général</b> , zone <b>Facteur d'échelle de sortie</b> Accès : Lecture/Ecriture
Parent	N'importe quelle vue hiérarchique dans Transformer Accès : Lecture
ParentCategories	Fenêtre des dimensions, fenêtre des catégories Accès : Lecture
Partition	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Numéro de partition</b> Accès : Lecture/Ecriture
Password	Feuille des propriétés du PowerCube, onglet <b>Sortie</b> , zone <b>Mot de passe</b>  Feuille des propriétés du code d'accès, onglet <b>Général</b> , zone <b>Mot de passe</b> Accès : écriture
Path	Explorateur Windows Accès : Lecture
PopulateByDataSource	Aucun Accès : Lecture/Ecriture

Propriétés	Equivalent IU
Position	Feuille des propriétés de la colonne, onglet <b>Général</b> , zone <b>Position</b> Accès : Lecture/Ecriture
PowerCubesPath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>PowerCubes</b> Accès : Lecture/Ecriture
Precision	Feuille des propriétés de la mesure, onglet <b>Format</b> , zone <b>Précision</b> Accès : Lecture/Ecriture
QualifiedName	Grille des dimensions Accès : Lecture
QuarterType	Assistant Dates Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Fonction de date</b> Accès : écriture
QyPath	Feuille des propriétés des préférences, onglet <b>Répertoires</b> , zone <b>Fichiers temporaires de modèles</b> Accès : Lecture
Rate	Feuille des propriétés de la table des devises Accès : Lecture/Ecriture
RefreshDescription	Feuille des propriétés du niveau, onglet <b>Source</b> , zone <b>Mise à jour</b> , case à cocher <b>Description</b> Accès : Lecture/Ecriture
RefreshLabel	Feuille des propriétés du niveau, onglet <b>Source</b> , zone <b>Mise à jour</b> , case à cocher <b>Libellé</b> Accès : Lecture/Ecriture
RefreshShortName	Feuille des propriétés du niveau, onglet <b>Source</b> , zone <b>Mise à jour</b> , case à cocher <b>Nom abrégé</b> Accès : Lecture/Ecriture



Propriétés	Equivalent IU
RegularRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> Accès : Lecture/Ecriture
RegularWeight	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Pondération ordinaire</b> Accès : Lecture/Ecriture
ReverseSign	Feuille des propriétés de la mesure, onglet <b>Général</b> , case à cocher <b>Inverser le signe</b> Accès : Lecture/Ecriture
Rollup	Feuille des propriétés de la catégorie spéciale, onglet <b>Général</b> , case à cocher <b>Cumul sur catégorie</b> Accès : Lecture/Ecriture
RollupTiming	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Délai ordinaire</b> (pour une mesure calculée) Accès : Lecture/Ecriture
RowsAsSample	Feuille des propriétés des préférences, onglet <b>Conception automatique</b> , zone <b>Nombre de lignes utilisées comme échantillon</b> Accès : Lecture/Ecriture
RowsChecked	Feuille des propriétés des préférences, onglet <b>Conception automatique</b> , zone <b>Nombre de lignes vérifiées</b> Accès : Lecture/Ecriture
RunningPeriods	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Nombre de périodes</b> (quand la date relative est personnalisée et que le type est un total cumulatif) Accès : Lecture/Ecriture
Server	Feuille des propriétés du PowerCube, onglet <b>Traitement</b> , zone <b>Traitement</b> Accès : Lecture/Ecriture
ServerModelPath	Feuille des propriétés du modèle, onglet <b>Serveur</b> , zone <b>Chemin d'accès du modèle</b> Accès : Lecture/Ecriture

Propriétés	Equivalent IU
ServerPath	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Fichier de données sur le serveur</b> (quand la source de données est le serveur)  Accès : Lecture/Ecriture
ServerQuery	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Type de source de données</b>  Accès : Lecture/Ecriture
ServicesBuildNumber	Zone <b>A propos de</b>  Accès : Lecture
ServicesVersionText	Zone <b>A propos de</b>  Accès : Lecture
SetsCurrentPeriod	Feuille des propriétés des sources de données, onglet Général, case à cocher Définit la période courante  Accès : Lecture/Ecriture
ShortName	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Nom abrégé</b>  Accès : Lecture/Ecriture
Signon	Feuille des propriétés du PowerCube, onglet <b>Sortie</b> , zone <b>Code d'accès</b> (quand la valeur du type de base de données est différente de Local)  Accès : Lecture/Ecriture
Signons	Liste des codes d'accès  Accès : Lecture
Size	Feuille des propriétés de la colonne, onglet <b>Général</b> , zone <b>Taille</b>  Accès : Lecture/Ecriture
Size (Model)	Explorateur Windows  Accès : Lecture

Propriétés	Equivalent IU
SortComparisonRule	Feuille des propriétés des préférences, onglet <b>Général</b> , zone <b>Règles de comparaison pour le tri</b> Accès : Lecture/Ecriture
SourceType	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Type de source de données</b> Accès : Lecture/Ecriture
SpecialCategoryCount	Grille des dimensions (deuxième chiffre entre crochets sous le nom de la dimension) Accès : Lecture
SQLExpression	Feuille des propriétés de la source de données, onglet <b>Affichage du code SQL</b> Accès : Lecture
Status	Feuille des propriétés du PowerCube, onglet <b>Traitement</b> , zone <b>Statut</b> Accès : Lecture
StorageType	Feuille des propriétés de la mesure, onglet <b>Général</b> , zone <b>Type de données</b> Accès : Lecture/Ecriture
SummaryLevel	Feuille des propriétés du PowerCube, onglet <b>Groupe</b> , zone <b>Récapituler toutes les catégories externes du niveau</b> Accès : Lecture/Ecriture
SuspendedModels	Menu <b>Fichier</b> , commande <b>Afficher les modèles en attente</b> , zone <b>Sélection de modèles en attente</b> Accès : Lecture
TargetLevel	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Période cible</b> (quand la date relative est personnalisée) Accès : Lecture/Ecriture
TargetOffset	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Décalage</b> (quand la date relative est personnalisée) Accès : Lecture/Ecriture

Propriétés	Equivalent IU
ThousandPoint	Feuille des propriétés de la source de données, onglet <b>Général</b> , zone <b>Séparateur de milliers</b> (pour les sources de données de fichier à plat)  Accès : Lecture/Ecriture
Time	Explorateur Windows  Accès : Lecture
TimeArrayColumn	Feuille des propriétés de la colonne, zone <b>Nom de la colonne</b>  Accès : Lecture/Ecriture
TimeArrayStartMonth	Feuille des propriétés de la colonne, onglet <b>Tableau</b> , zone <b>Mois de début</b>  Accès : Lecture/Ecriture
TimeArrayType	Feuille des propriétés de la colonne, onglet <b>Tableau</b> , zone <b>Type de tableau</b>  Accès : Lecture/Ecriture
TimeRank	Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Rang des niveaux de temps</b>  Accès : Lecture/Ecriture
TimeStateRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Cumul d'état temporel</b>  Accès : Lecture/Ecriture
TimeStateWeight	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Pondération d'état temporel</b>  Accès : Lecture/Ecriture
ToDateLevel	Feuille des propriétés de la catégorie spéciale, onglet <b>Date relative</b> , zone <b>Période de base</b> (quand la date relative est personnalisée)  Accès : Lecture/Ecriture
Type	Aucun  Accès : Lecture

Propriétés	Equivalent IU
Unique	Feuille des propriétés du niveau, onglet <b>Source</b> , case à cocher <b>Unique</b> Accès : Lecture/Ecriture
UniqueMove	Feuille des propriétés du niveau, onglet <b>Source</b> , case à cocher <b>Déplacer</b> Accès : Lecture/Ecriture
UserID	Feuille des propriétés du code d'accès, onglet <b>Général</b> , zone <b>ID utilisateur</b> Accès : Lecture/Ecriture
Version	Menu ?, commande A propos d' IBM Cognos PowerPlay Transformer, zone A propos de Transformer Accès : Lecture
Views	Fenêtre des dimensions, onglet <b>Dimension</b> ou <b>Classe d'utilisateurs</b> Accès : Lecture
ViewStatus	Menu <b>Diagramme</b> , commande <b>Retrait</b> , <b>Exclusion</b> , <b>Omission</b> , <b>Récapitulatif</b> ou <b>Sommet</b> (quand une catégorie est sélectionnée dans une vue) Accès : Lecture/Ecriture
ViewType	Feuille des propriétés du PowerCube, onglet <b>Dimensions</b> (quand vous cliquez avec le bouton droit de la souris sur une sélection) Accès : Lecture/Ecriture
WeekAdd	Assistant Dates Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Ajouter une semaine supplémentaire</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDrillDown)
WeekSpan	Assistant Dates Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Semaines partielles</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDrillDown)

Propriétés	Equivalent IU
WeekStart	Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Premier jour de la semaine</b> Accès : Lecture/Ecriture
WeekStartDay	Assistant Dates Accès : écriture
WorkingDay	Assistant Dates Feuille des propriétés de la dimension (Date), onglet <b>Date</b> , zone <b>Jours de la semaine</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDimension)
WorkingDays	Assistant Dates Feuille des propriétés de la dimension (Date), onglet <b>Date</b> , zone <b>Jours de la semaine</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDimension)
YearStartDay	Assistant Dates Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Début de l'année</b> Accès : écriture (Assistant Dates) lecture/écriture (DateDrillDown)
YearType	Assistant Dates Feuille des propriétés du niveau (Date), onglet <b>Date</b> , zone <b>Fonction de date</b> Accès : écriture

## Listes des valeurs et constantes

La table suivante présente un récapitulatif des équivalences de l'interface utilisateur pour les listes de valeurs et les constantes.

Constantes	Equivalent IU
xtrAllocationType	Feuille des propriétés du niveau, onglet <b>Allocation</b> , menu contextuel (quand vous cliquez avec le bouton droit de la souris sur une mesure sélectionnée)

Constantes	Equivalent IU
xtrAssociationRole	<p>Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)</p> <p>Feuille des propriétés du niveau, onglet <b>Source</b> ou onglet <b>Classement</b></p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b>)</p> <p>Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée)</p>
xtrAssociationType	<p>Feuille des propriétés de la dimension, onglet Général, zone Colonne de cumul externe (quand la case à cocher "Contient des valeurs de mesure avec cumul calculé à l'externe" est activée dans l'onglet "Général" de la feuille des propriétés de la source de données)</p> <p>Feuille des propriétés du niveau des données, onglet <b>Source</b> ou</p> <p>Feuille des propriétés de la mesure, onglet <b>Type</b> (quand l'option <b>Type</b> a pour valeur <b>Colonne</b>) ou</p> <p>Feuille des propriétés de la table des devises, zone Colonnes de la table de base (quand la case à cocher "Utiliser une source de données des devises externe" est activée)</p>
xtrCharacterType	<p>Feuille des propriétés de la source de données (source de données de fichier à plat), zone <b>Jeu de caractères</b></p>
xtrCubeConsolidate	<p>Feuille des propriétés du PowerCube, onglet <b>Général</b>, zone <b>Consolidation</b></p>
xtrCubeCreation	<p>Feuille des propriétés du PowerCube, onglet <b>Traitement</b>, zone <b>Création de cube</b></p>
xtrCubeOptimize	<p>Feuille des propriétés du PowerCube, onglet <b>Traitement</b>, zone <b>Optimisation</b></p>
xtrCubeStatus	<p>Feuille des propriétés du PowerCube, onglet <b>Traitement</b>, zone <b>Statut</b></p>

Constantes	Equivalent IU
xtrCurrencyTableType	Feuille des propriétés de la table des devises
xtrDataClass	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Classe de données</b>
xtrDateCategoriesGeneration	Feuille des propriétés du niveau (Date), onglet <b>Date</b> , case à cocher <b>Générer toutes les catégories de la période</b>
xtrDateFormat	Feuille des propriétés de la colonne, onglet <b>Date</b> , zone <b>Format de date</b>
xtrDateLevel	Feuille des propriétés de la colonne, onglet <b>Date</b> , zone <b>Degré de détail</b>
xtrDuplicateRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Cumul des doubles</b>
xtrGenerateOptions	Feuille des propriétés du niveau (Date), onglet <b>Général</b> , zone <b>Inclusion</b>
xtrInclusion	Feuille des propriétés de l'objet, onglet <b>Général</b> , zone <b>Inclusion</b>
xtrMeasureType	Feuille des propriétés de la mesure, onglet <b>Type</b>
xtrMissingValue	Feuille des propriétés de la mesure, onglet <b>Général</b> , zone <b>Valeurs manquantes</b>
xtrObjectType	Élément de l'interface utilisateur qui référence un objet OLE de Transformer
xtrOrigin	Aucun
xtrPowerCubeGeneration	Feuille des propriétés de la source de données, onglet <b>Général</b> , zone <b>lecture des données sources</b> , case à cocher <b>Au moment de la création du PowerCube</b> (cases d'option)
xtrPreferences	Menu <b>Fichier</b> , commande <b>Préférences</b> , feuille des propriétés des préférences
xtrRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Cumul ordinaire</b>
xtrRollupTiming	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Délai ordinaire</b> (pour une mesure calculée)



Constantes	Equivalent IU
xtrSourceType	Feuille des propriétés de la source de données, onglet <b>Source</b> , zone <b>Type de source de données</b>
xtrSpecialFunction	Feuille des propriétés du niveau de date, onglet <b>Date</b> , zone <b>Fonction de date</b>
xtrStorage	Feuille des propriétés de la mesure, onglet <b>Général</b> , zone <b>Type de données</b>
xtrTimeAggregate	Feuille des propriétés de la catégorie spéciale, onglet Date relative, zone Type (quand la date relative est personnalisée)
xtrTimeArrayType	Feuille des propriétés de la colonne, onglet <b>Tableau</b> , zone <b>Type de tableau</b>
xtrTimeRollup	Feuille des propriétés de la mesure, onglet <b>Cumul</b> , zone <b>Cumul d'état temporel</b>
xtrTimeType	Assistant Dates
xtrViewStatus	Menu Diagramme, sélectionner l'une des commandes suivantes : Retrait, Exclusion, Omission, Récapitulatif ou Sommet (quand une catégorie est sélectionnée dans une vue)
xtrViewType	Feuille des propriétés du PowerCube, onglet <b>Dimension</b> , menu contextuel (quand vous cliquez avec le bouton droit de la souris sur une dimension sélectionnée)
xtrWeekAdd	Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Ajouter une semaine supplémentaire</b>
xtrWeekDay	Feuille des propriétés de la dimension (Date), onglet <b>Date</b> , zone <b>Jours de la semaine</b>  Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Premier jour de la semaine</b>
xtrWeekSpan	Feuille des propriétés de la catégorie de branche (Date), onglet <b>Date</b> , zone <b>Semaines partielles</b>



---

## Chapitre 9. Exemples

Cette section offre un large éventail d'exemples de codes Visual Basic illustrant certains concepts d'automatisation OLE. Les exemples de codes ont été testés dans Visual Basic 2008.

**Remarque :** Si vous le souhaitez, vous pouvez réutiliser les exemples fournis dans cette section. Notez qu'une telle réutilisation peut impliquer davantage que juste copier et coller l'exemple de code. Il est notamment possible que votre environnement de développement traite les sauts de ligne différemment.

Vous pouvez également utiliser les exemples fournis avec l'installation du produit. Les packs d'exemples Entrepôt de données VA (analyse) et Entrepôt de données VA (requête) se trouvent dans le fichier IBM\_Cognos\_Samples.zip, dans le répertoire *installation\_location\webcontent\samples\content\*.

Pour plus d'informations sur l'installation et la configuration des exemples IBM Cognos, voir le *Guide d'installation et de configuration* de votre produit.

---

### Exemple d'ouverture d'un modèle et de spécification d'un ordre par association

L'exemple suivant ouvre un modèle et spécifie un ordre en associant une branche de niveaux à un niveau de convergence.

```
Sub Example1()  
    Dim objTransApp As Object  
    Dim objModel As Object  
    Dim objDrill As Object  
    Dim objAssociation As Object  
    Dim objColumn As Object  
    Dim strIBMCognos10Location As String  
    Dim strModelPath As String  
    Dim strModelSource As String  
    'Change these paths to match your installation  
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\  
    strModelSource = "Great outdoors 8.mdl"  
    strModelPath = strIBMCognos10Location & _  
        "webcontent\samples\models\Transformer8\EN\" & strModelSource  
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")  
    With objTransApp  
        .DataSourcePath = strIBMCognos10Location & "bin"  
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"  
    End With  
    objModel = objTransApp.OpenModel(strModelPath)  
    objDrill = objModel.Dimensions("Retailers").DrillDowns(2)  
    'Create an OrderBy association for the convergence level.  
    objAssociation = objDrill.ConvergenceLevel.Associations.Add()  
    'Select the drill-down path to which the sort applies.  
    objAssociation.Context = objDrill  
    'Select the data source column, RetailerName, on which to base  
the sort.  
    objColumn = objModel.DataSources("Retailer site (csv)").Columns("Retailer name")  
    With objAssociation  
        .AssociationRole = xtrAssociationRole.trAssociationOrderBy  
        .AssociationType = xtrAssociationType.trAssociationQuery  
        .Label = objColumn.Name  
        .Update()  
    End With  
End Sub
```

```

End With
objDrill.Levels("Retailer site").OrderByDescending(objAssociation.Context) =
False
With objModel
    .GenerateCategories()
    .Update()
    .SaveAs("Great outdoors 8X.mdl")
    .Close()
End With
objColumn = Nothing
objAssociation = Nothing
objDrill = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple d'ouverture d'un modèle et d'ajout d'un calcul

L'exemple suivant ouvre un modèle et ajoute une définition de calcul à la dimension Produits. L'expression utilise la fonction de partage et comprend un ensemble de catégories permettant de créer deux catégories calculées.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie "parentCategory =" ou "childCategory =", vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte." Veuillez utiliser plutôt l'interface utilisateur.

```

Sub Example2()
    Dim objTransApp As Object
    Dim model As Model
    Dim dimension As Dimension
    Dim calcDef As CalculationDefinition
    Dim catSet As CategorySet
    Dim drillDown As DrillDown
    Dim parentCategory As Category
    Dim childCategory As Category
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    model = objTransApp.OpenModel(strModelPath)
    dimension = model.Dimensions.Item("Products")
    calcDef = dimension.CalculationDefinitions.Add()
    catSet = calcDef.CategorySets.Add()

    drillDown = dimension.DrillDowns.Item(1)
    parentCategory = drillDown.Categories.Item(1) 'select "Camping
Equipment"
    For child_category_index = 1 To parentCategory.ChildCategories.Count
        childCategory = parentCategory.ChildCategories.Item(child_category_index)
        If childCategory.Name = "Cooking Gear" Or childCategory.Name
= "Tents" Then
            catSet.Categories.Add(childCategory)
        End If
    Next
    catSet.Label = "Set 1"

```

```

With calcDef
    .ExpressionText = "share ( catset ( ""Set 1"" ) , "" &
-
        parentCategory.Code & """" )"
    .Group = False
    .Name = "share(""Set 1"", ""Camping Equipment"")"
    .Update()
End With
With model
    .SaveAs("great outdoors salesX.mdl")
    .Close()
End With
End Sub

```

---

## Exemple de création d'une catégorie de date relative

L'exemple suivant utilise l'objet SpecialCategory pour créer une catégorie de date relative, puis définit les propriétés applicables.

```

Sub Example3()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objSpecCategory As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Sales and Marketing.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    'Create a relative time special category in a time dimension.
    objSpecCategory = _
        objModel.Dimensions("Time").Categories.Add(xtrObjectType.trSpecialCategory)
    With objSpecCategory
        .Aggregate = xtrTimeAggregate.trAggregateRunning
        .Code = "Five Month Period"
        .ContextLevel = "Quarter"
        .ContextOffset = -3
        .Name = "Five Month Period"
        .Rollup = True
        .RunningPeriods = 5
        .TargetLevel = "Month"
        .TargetOffset = -1
        .Update()
    End With
    With objModel
        .SaveAs("Sales and MarketingX.mdl")
        .Close()
    End With
    objSpecCategory = Nothing
    objModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout d'un groupe de cubes

L'exemple de macro suivant ajoute un groupe de cubes à un modèle existant en utilisant la dimension "Région de vente" comme base du groupe. Toutefois, il ne crée pas de cube enfant dans le groupe.

```
Sub Example4()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDimRegion As Object
    Dim objCubesByRegion As Object
    Dim strCategoryCode As String
    Dim objChildCube As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    objDimRegion = objModel.Dimensions("Sales region")
    objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)
    With objCubesByRegion
        'Specify the category levels on which to base the cubes
        in the group.
        .GroupDimension = objDimRegion
        .GroupLevel = objDimRegion.DrillDowns(1).Levels("Sales region")
        'Specify the level of detail and summary level.
        .DetailLevel = objDimRegion.DrillDowns(1).Levels("Country")
        .SummaryLevel = objDimRegion.DrillDowns(1).Levels("Sales
region")
        .Name = "Regions"
        .MDCFile = ""
        .CubeCreation = xtrCubeCreation.trCubeCreationON
        .Optimize = xtrCubeOptimize.trOptimizeDefault
        .CompressMDC = False
        .CacheCrossTabs = False
        .MeasureInclude(objModel.Measures("Unit cost")) = False
        .MeasureName = "Revenue Made"
        .Update()
    End With
    'Name each cube in the group after its category name
    For index = 1 To objCubesByRegion.GroupLevel.CategoryCount
        strCategoryCode = objCubesByRegion.GroupLevel.LevelCategories(index).Code
        objChildCube = objCubesByRegion.ChildCubes(strCategoryCode)
        With objChildCube
            .Name = objCubesByRegion.GroupLevel.LevelCategories(index).Name
            .Update()
        End With
    Next
    'Do not generate a child cube for Central Europe.
    objChildCube = objCubesByRegion.ChildCubes("Central Europe")
    With objChildCube
        .CubeCreation = xtrCubeCreation.trCubeCreationOFF
        .Update()
    End With
    'Generate the cubes (this may take a few minutes)
    objCubesByRegion.CreateMDCFile()
    With objModel
```

```

        .SaveAs("great outdoors salesX.mdl")
        .Close()
    End With
    objChildCube = Nothing
    objDimRegion = Nothing
    objCubesByRegion = Nothing
    objModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout d'une source de données supplémentaire à un modèle

L'exemple suivant ajoute une autre source de données à un modèle, puis définit l'allocation à l'aide de la nouvelle source de données. Il crée la mesure Prévisions et alloue proportionnellement la mesure Revenus en tant que facteur de pondération.

```

Sub Example5()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDataSource As Object
    Dim objDimension As Object
    Dim objColumn As Object
    Dim objLevel As Object
    Dim objMeasure As Object
    Dim objByMeasure As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strDataSource = "prod_plan.csv"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\great
outdoors sales\" & _
        strDataSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    'Add and define a second datasource.
    objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
    With objDataSource
        .CharacterType = xtrCharacterType.trCharAnsiSingleByte
        .External = False
        .FieldSeparator = ","
        .LocalPath = strDataPath
        .MaximizeSpeed = True
        .Name = "Product Plan (csv)"
        .SourceType = xtrSourceType.trFlatFileColumnNames
        .Update()
    End With
    'Specify data classes and rename columns.
    objColumn = objDataSource.Columns(1)
    With objColumn
        .DataClass = xtrDataClass.trDataClassDate
        .Name = "Order Date"
        .Update()
    End With

```

```

objColumn = objDataSource.Columns(3)
With objColumn
    .DataClass = xtrDataClass.trDataClassDescription
    .Name = "Product line code"
    .Update()
End With
'Add a measure.
objColumn = objDataSource.Columns("EXPECTED_VOLUME")
With objColumn
    .DataClass = xtrDataClass.trDataClassQuantity
    .InputScale = 0
    .Name = "Forecast"
    .Update()
End With
objMeasure = objModel.Measures.Add()
With objMeasure
    .AssociateWith(objColumn, xtrAssociationRole.trAssociationSource)
    .Description = "Forecasted volume for product line."
    .AllowDrillThrough = False
    .Update()
End With
'Test whether allocation is possible.
objLevel = objModel.Dimensions("Products").DimensionLevels(1)
If objLevel.CanAllocate = True Then
    objMeasure = objModel.Measures("Forecast")
    'Test whether the specified measure can be allocated.
    If objLevel.CanAllocateMeasure(objMeasure) = True Then
        'Test whether allocation by proportion is already set.
        If objLevel.AllocationType(objMeasure) <> _
            xtrAllocationType.trAllocationByAnotherMeasure Then
            objByMeasure = objModel.Measures("Revenue")
            'Test whether the specified measure can be used
            as a weighting factor.
            If objLevel.CanAllocateByMeasure(objByMeasure) = True
            Then
                objLevel.SetAllocation(objMeasure, _
                    xtrAllocationType.trAllocationByAnotherMeasure,
                    objByMeasure)
            End If
        End If
    End If
End If
'Suppress allocation to the Margin range dimension.
objDimension = objModel.Dimensions("Margin range")
If objDimension.CanAllocate = True Then
    objDimension.SetAllocation(objMeasure, xtrAllocationType.trAllocationNA)
End If
With objModel
    .SaveAs("great outdoors salesX.mdl")
    .Close()
End With
objByMeasure = Nothing
objMeasure = Nothing
objLevel = Nothing
objColumn = Nothing

```

6

---

## Exemple d'ouverture d'un modèle et de modification des propriétés du cube

L'exemple suivant ouvre un modèle et modifie les propriétés du cube.

```

Sub Example6()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objCube As Object

```



```

Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objCube = objModel.Cubes.Item(1)
With objCube
    .BlockParentTotals = True
    .CacheCrossTabs = True
    .CompressMDC = False
    .Consolidate = xtrCubeConsolidate.trConsolidateDefault
    .CubeCreation = xtrCubeCreation.trCubeCreationON
    .DesiredPartitionSize = 500000
    .EstimatedRows = 10000000
    .IncrementalUpdate = False

    .MaxNumPartLevels = 5
    .MDCFile = "GoCube"
    .Optimize = xtrCubeOptimize.trOptimizeAutoPartition
    .Name = "Great Outdoors Sales (Optimized)"
    .Update()
    .CreateMDCFile()
End With
objModel.SaveAs("great outdoors salesX.mdl")
objModel.Close()
objCube = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une vue personnalisée

L'exemple ci-après permet de créer une vue personnalisée. Il associe ensuite cette vue personnalisée au cube.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie "objCategory =", vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte". Veuillez utiliser plutôt l'interface utilisateur.

```

Sub Example7()
    Dim objTransApp As Object
    Dim model As Model
    Dim dimension As Dimension
    Dim measure As Measure
    Dim custom_view As CustomView
    Dim view As TransformerSDKLib.View
    Dim category As Category
    Dim cube As Cube
    Dim intX As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Sales and Marketing.mdl"

```

```

strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
model = objTransApp.OpenModel(strModelPath)
'Set a custom view for a dimension.
dimension = model.Dimensions.Item("Retailers")
'Exclude the "Planned revenue" measure
measure = model.Measures.Item("Planned revenue")
'Create custom view
custom_view = model.CustomViews.Add()
With custom_view
    .Name = "Central Europe"
    .DimensionInclude(dimension) = xtrViewType.trViewTypeCustom
    .MeasureInclude(measure) = False
    .Update()
End With
'Get associated View object
view = custom_view.DimensionView(dimension)
'Set a view for each category in the top level of the collection.
For intX = 1 To dimension.DrillDowns.Item(1).Categories.Count
    category = dimension.DrillDowns.Item(1).Categories(intX)
    If category.Name <> "Central Europe" Then
        view.SetViewStatus(category, xtrViewStatus.trViewStatusSummaryMom)
    End If
Next intX
'Associate a custom view with a cube
cube = model.Cubes.Item("Sales and Marketing")
cube.CubeCustomViews.Add(custom_view)
With model
    .SaveAs("Sales and MarketingX.mdl")
    .Close()
End With
objTransApp = Nothing
End Sub

```

---

## Exemple d'ouverture d'un modèle et d'ajout d'un enregistrement de devise

L'exemple suivant ouvre un modèle et ajoute un enregistrement de devise à une table des devises existante. Il définit aussi les taux de conversion du nouvel enregistrement.

```

Sub Example8()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objBaseTable As Object
    Dim objCurrencyRecord As Object
    Dim objCurrencyRate As Object
    Dim objDateDrillDown As Object
    Dim objLevel As Object
    Dim intX As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Great outdoors 8.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp

```

```

        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    objBaseTable = objModel.CurrencyTables.Add()
    With objBaseTable
        .CurrencyTableType = xtrCurrencyTableType.trCurrencyTableBase
        .Update()
    End With
    objDateDrillDown = objModel.Dimensions("Years").DrillDowns(1)
    objCurrencyRecord = objModel.CurrencyRecords.Add()
    objLevel = objDateDrillDown.Levels("Year")
    objCurrencyRecord.DateLevel = objLevel
    With objCurrencyRecord
        .CountryCode = "AUS"
        .Label = "Australian Dollar"
        .CurrencyDecimals = 2
        .CurrencyFormatOverride = True
        .CurrencyIsEMU = False
        .CurrencyIsEuro = False
        .CurrencySymbol = "$"
        .Update()
    End With
    For intX = 1 To objLevel.CategoryCount
        objCurrencyRate = objCurrencyRecord.CurrencyRates(intX)
        If objCurrencyRate.PopulateByDataSource = False Then
            Select Case objLevel.LevelDrillDowns(1).Categories(intX).KeyName
                Case "2004"
                    objCurrencyRate.Rate = 1.54
                Case "2005"
                    objCurrencyRate.Rate = 1.55
                Case Else
                    '
            End Select
            Select Case objCurrencyRate.Category.KeyName
                Case "2006"
                    objCurrencyRate.Rate = 1.56
                Case "2007"
                    objCurrencyRate.Rate = 1.57
                Case Else
                    '
            End Select
            objCurrencyRate.Update()
        End If
    Next intX
    With objModel
        .SaveAs("Great outdoors 8X.mdl")
        .Close()
    End With
    objLevel = Nothing
    objDateDrillDown = Nothing
    objCurrencyRate = Nothing
    objCurrencyRecord = Nothing
    objModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple de création d'un cube à l'aide des méthodes DoAutoDesign et TestBuild

L'exemple suivant crée un modèle et ajoute une source de données. Il utilise ensuite les méthodes DoAutoDesign et TestBuild pour créer un petit cube.

```

Sub Example9()
    Dim objTransApp As Object
    Dim objModel As Object

```

```

Dim objDataSource As Object
Dim strIBMCognos10Location As String
Dim strDataSource As String
Dim strDataPath As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strDataSource = "NATIONAL.ASC"
strDataPath = strIBMCognos10Location & _
    "webcontent\samples\datasources\cubes\PowerCubes\EN\National\"
& _
    strDataSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
objModel = objTransApp.NewModel
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
With objDataSource
    .GenerateCategories = True
    .GeneratePowerCube = xtrPowerCubeGeneration.trGenerationDefault
    .CharacterType = xtrCharacterType.trCharAnsiSingleByte
    .FieldSeparator = ","
    .LocalPath = strDataPath
    .MaximizeSpeed = True
    .Name = "National"
    .SetsCurrentPeriod = True
    .SourceType = xtrSourceType.trFlatFileColumnNames
    .Update()
End With
With objModel
    .Name = "National"
    .DoAutoDesign()
    .TestBuild(20, True)
    .Update()
    .SaveAs("NationalX.mdl")
    .Close()
End With
objDataSource = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de sélection, de modification et de mise à jour d'une dimension

L'exemple suivant sélectionne un objet Dimension dans la collection Dimensions, modifie une propriété et met la dimension à jour.

```

Sub Example10()
Dim objTransApp As Object
Dim objModel As Object
Dim objDimensions As Object
Dim objLocationsDim As Object
Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objDimensions = objModel.Dimensions
objLocationsDim = objDimensions.Item(3)

```

```

With objLocationsDim
    .ExcludeAutoPartition = True
    .Update()
End With
objModel.SaveAs("great outdoors salesX.mdl")
objModel.Close()
objLocationsDim = Nothing
objDimensions = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de suppression d'un niveau à partir d'une collection de niveaux

L'exemple suivant supprime un niveau de la collection Levels.

```

Sub Example11()
Dim objTransApp As Object
Dim objModel As Object
Dim objTimeDimension As Object
Dim objLevel As Object
Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "Sales and Marketing.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objTimeDimension = objModel.Dimensions.Item("Time")
objLevel = objTimeDimension.DrillDowns.Item(1).Levels.Item("Month")
objLevel.Delete()
objTimeDimension.Update()
objModel.SaveAs("Sales and MarketingX.mdl")
objModel.Close()
objLevel = Nothing
objTimeDimension = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de déplacement d'un objet de mesure et de modification du cumul de la mesure Revenus

L'exemple suivant déplace un objet Measure vers la première position de la collection Measures. Il remplace le cumul de la mesure Revenus par un cumul en double. La mesure Revenus utilise la mesure Quantité pour créer une moyenne pondérée.

```

Sub Example12()
Dim objTransApp As Object
Dim objModel As Object
Dim objMeasures As Object
Dim currentMeasure As Measure
Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation

```

```

strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objMeasures = objModel.Measures
objMeasures.Item(2).Move(1)
currentMeasure = objMeasures("Revenue")
With currentMeasure
    .RegularRollup = xtrRollup.trRollupAverage
    .RegularWeight = objModel.Measures("Quantity").Name
    .RollupTiming = xtrRollupTiming.trTimingDefault
    .DuplicateRollup = xtrDuplicateRollup.trDuplicateRollupAverage
    .DuplicateWeight = objModel.Measures("Quantity").Name
    .Format = "$#,##0"
    .FormatDecimals = 2
    .IgnoreMissingValue = False
    .MissingValue = xtrMissingValue.trMissingValueZERO
    .ReverseSign = False
    .ShortName = "Revenue"
    .Update()
End With
objModel.SaveAs("great outdoors salesX.mdl")
objModel.Close()
objMeasures = Nothing
objModel = Nothing

```

---

## Exemple de création d'une partition et de vérification du modèle

L'exemple suivant ouvre un modèle, crée une partition pour l'Europe centrale et les catégories descendantes, puis vérifie le modèle.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie "objCategory =", vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte". Veuillez utiliser plutôt l'interface utilisateur.

```

Sub Example13()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objCategory As Object
    Dim intX As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    objModel.ResetPartitions()
    objCategory = objModel.Dimensions("Sales region").Drilldowns(1).Categories(2)
    objCategory.Partition = 1

```

```

objCategory.Update()
For intX = 1 To objModel.CheckModel.Count
    MsgBox(objModel.CheckModel(intX).Name)
Next intX
With objModel
    .SaveAs("great outdoors salesX.mdl")
    .Close()
End With
objCategory = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple d'ouverture d'un modèle et d'accès au détail du PowerCube

L'exemple suivant ouvre un modèle et utilise la mesure Quantité pour accéder au détail d'un PowerCube.

```

Sub Example14()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objMeasure As Object
    Dim objReport As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    Dim strReportPath As String
    Dim strReportSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strReportSource = "drill_through_to_cube_7.mdc"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    strReportPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\" &
strReportSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    objMeasure = objModel.Measures.Item("Quantity")
    With objMeasure
        .AllowDrillThrough = True
        .Update()
    End With
    objReport = objMeasure.DrillThroughTargets.Add(strReportPath,
"Default Report")
    objModel.SaveAs("great outdoors salesX.mdl")
    objModel.Close()
    objReport = Nothing
    objMeasure = Nothing
    objModel = Nothing

```

---

## Exemple d'ajout du rôle Auteurs à une vue personnalisée

L'exemple suivant utilise l'espace noms **Cognos** pour ajouter le rôle Auteurs à une vue personnalisée.

```

Sub Example15()
    Dim objTransApp As Object
    Dim model As Model
    Dim new_namespace As TransformerSDKLib.Namespace
    Dim securityObject As SecurityObject

```

```

Dim customView As CustomView
Dim CAMID_of_Namespace As String
Dim CAMID_of_User As String
Dim CAMID_of_Object As String
Dim Name_of_Namespace As String
Dim Name_of_User As String
Dim Name_of_Object As String
Dim ID_of_Namespace As String
Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
Name_of_Namespace = "Cognos"
ID_of_Namespace = ""
Name_of_User = ""
Name_of_Object = "Authors"
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location &
"bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
model = objTransApp.OpenModel(strModelPath)
'Provide namespace, username, and password in that order
objTransApp.Logon("Cognos", "", "") 'Log into the Cognos namespace
using Anonymous user
new_namespace = model.Namespaces.Add()
'Provide: Namespace and Object names
'Receive: Both Namespace's and Object's CAMIDs
With new_namespace
    .Name = Name_of_Namespace
    .ObjectName = Name_of_Object 'Authors is a group
    .Update()
    CAMID_of_Namespace = .CAMID
    CAMID_of_Object = .ObjectCAMID
    CAMID_of_User = ""
End With
'Another way to get CAMIDs below (Commented Out)
'The code below can be adapted to use any Namespace and User
'Provide: Namespace ID and user
'Receive: Namespace and User CAMIDs and Namespace name
'With new_namespace
'    .ID = ID_of_Namespace
'    .User = Name_of_User
'    .Update()
'    CAMID_of_Namespace = .CAMID
'    CAMID_of_User = .UserCAMID
'    Name_of_Namespace = .Name
'End With
'Create custom view
customView = model.CustomViews.Add()
With customView
    .Name = Name_of_Object
    .Update()
End With
'Create and add security object to custom view
securityObject = new_namespace.SecurityObjects.Add()
With securityObject
    .Name = CAMID_of_Object 'provide the User or Object CAMID
here
    .DisplayName = Name_of_Object
    .Type = xtrSecurityType.trSecurityType_Role
    .AddToCustomView(customView)

```



```

        .Update()
    End With
    With model
        .SaveAs("great outdoors salesX.mdl")
        .Close()
    End With
    objTransApp.Logoff()
    objTransApp = Nothing
End Sub

```

---

## Exemple de recherche d'un modèle en attente

L'exemple suivant recherche un modèle en attente et, s'il en trouve un, affiche un message si le modèle est corrompu. Il utilise ensuite la méthode `RemoveSuspendedModel` pour le supprimer de la collection `SuspendedModels`.

```

Sub Example16()
    Dim objTransApp As Object
    Dim objSuspendedModel As Object
    Dim intX As Integer
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    If objTransApp.SuspendedModels.Count = 0 Then
        MsgBox("There are no suspended models")
    Else
        For intX = objTransApp.SuspendedModels.Count To 1 Step -1
            objSuspendedModel = objTransApp.SuspendedModels(intX)
            If objSuspendedModel.IsBad = True Then
                MsgBox("Model: " + objSuspendedModel.ModelName +
                    -
                    ", located at " + objSuspendedModel.QyPath + " is
                    corrupt")
            End If
            objTransApp.RemoveSuspendedModel(objSuspendedModel)
        Next intX
    End If
    objSuspendedModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple d'ouverture d'un modèle et de création d'une vue des dimensions

L'exemple suivant ouvre un modèle et crée une vue des dimensions pour la dimension `Détaillants`. La vue fournit une branche de niveaux complète pour l'Europe centrale, mais seulement le détail du niveau récapitulatif pour les autres régions. Ensuite, il associe la vue des dimensions à un cube.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie `objCategory =`, vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte". Veuillez utiliser plutôt l'interface utilisateur.

```

Sub Example17()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDimension As Object
    Dim objView As Object
    Dim objCategory As Object
    Dim objCube As Object
    Dim intX As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String

```

```

'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "Sales and Marketing.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objDimension = objModel.Dimensions("Retailers")
objCube = objModel.Cubes("Sales and Marketing")
If objCube.DimensionViewType(objDimension) = _
xtrViewType.trViewTypeAllCategories Then
    objView = objDimension.Views.Add()
    With objView
        .Name = "Central Europe"
        .ViewType = xtrViewType.trViewTypeCustom
        .Update()
        .Parent.Update()
    End With
    For intX = 1 To objDimension.DrillDowns(1).Categories.Count
        objCategory = objDimension.DrillDowns(1).Categories(intX)
        If objCategory.Name <> "Central Europe" Then
            objView.SetViewStatus(objCategory,
xtrViewStatus.trViewStatusSummaryMom)
            If objView.GetViewStatus(objCategory) <> _
xtrViewStatus.trViewStatusSummaryMom Then
                MsgBox("Failed to set ViewStatus!")
            End If
        End If
    Next intX
    objCube.DimensionView(objDimension) = objView
    objCube.Update()
End If
With objModel
    .SaveAs("Sales and MarketingX.mdl")
    .Close()
End With
objCube = Nothing
objCategory = Nothing
objView = Nothing
objDimension = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout d'un groupe de cubes à un modèle

Cette macro ajoute un groupe de cubes à un modèle existant. La dimension "Région de vente" devient la base du groupe.

```

Sub Example18()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDataSource As Object
    Dim objRegionsDrill As Object
    Dim objCubesByRegion As Object
    Dim objChildCube As Object
    Dim strCategoryCode As String
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"

```

```

strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objRegionsDrill = objModel.Dimensions("Sales region").DrillDowns(1)
objCubesByRegion = objModel.Cubes.Add(xtrObjectType.trCubeGroup)
With objCubesByRegion
    .GroupDimension = objModel.Dimensions("Sales region")
    .GroupLevel = objRegionsDrill.Levels("Sales region")
    .DetailLevel = objRegionsDrill.Levels("Branch")
    .SummaryLevel = objRegionsDrill.Levels("Sales region")
    .CacheCrossTabs = False
    .CompressMDC = False
    .CubeCreation = xtrCubeCreation.trCubeCreationON
    .MeasureInclude(objModel.Measures("Product cost")) = False
    .Name = "Sales Regions"
    .Optimize = xtrCubeOptimize.trOptimizeDefault
    .Update()
End With
'Name each cube in the group after its category name
For index = 1 To objCubesByRegion.GroupLevel.CategoryCount
    strCategoryCode = objCubesByRegion.GroupLevel.LevelCategories(index).Code
    objChildCube = objCubesByRegion.ChildCubes(strCategoryCode)
    With objChildCube
        .Name = objCubesByRegion.GroupLevel.LevelCategories(index).Name
        .Update()
    End With
Next
With objModel
    .SaveAs("great outdoors salesX.mdl")
    .Close()
End With
objChildCube = Nothing
objCubesByRegion = Nothing
objRegionsDrill = Nothing
objDataSource = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'un modèle et de mise à jour des propriétés pour une dimension de date

L'exemple de macro suivant crée un modèle et met à jour plusieurs propriétés associées à la dimension de date.

```

Sub Example19()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDataSource As Object
    Dim objDateDim As Object
    Dim objDateLevel As Object
    Dim strIBMCognos10Location As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strDataSource = "NATIONAL.ASC"
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\National\"
    & _

```

```

        strDataSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
objModel = objTransApp.NewModel
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
With objDataSource
    .LocalPath = strDataPath
    .Name = "National"
    .SetsCurrentPeriod = True
    .SourceType = xtrSourceType.trFlatFileColumnNames
    .Update()
End With
With objModel
    .Name = "National"
    .DoAutoDesign()
    .Update()
End With
objDateDim = objModel.Dimensions("Date")
With objDateDim
    .EarliestDate = 19900101
    .GenerateTimePeriod(xtrTimeType.trTimeTypeMonth) = _
        xtrGenerateOptions.trGenerateAll
    .LatestDate = 20101231
    .GenerateDateCategories(19900101, 20101231)
    .WorkingDay(xtrWeekDay.trSunday) = False
    .ManualCurrentPeriod = False
    .Update()
End With
objDateLevel = objDateDim.DimensionLevels("Month")
With objDateLevel
    .DateFormat = "MMMM, yyyy"
    .GenerateDateCategories = xtrDateCategoriesGeneration.trGenerateDatesAll
    .Update()
End With
With objModel
    .TestBuild(20, True)
    .GenerateCategories()
    MsgBox("Current Period:" + objDateDim.CurrentPeriod.Name)
    .SaveAs("NationalX.mdl")
    .Close()
End With
objDateLevel = Nothing
objDateDim = Nothing
objDataSource = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une branche de niveaux secondaire

L'exemple suivant crée une branche de niveaux secondaire dans une dimension.

```

Sub Example20()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objLocationsDim As Object
    Dim objLevel As Object
    Dim objAltDrill As Object
    Dim objNewLevel As Object
    Dim objRefSource As Object
    Dim objRefLabel As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Great outdoors 8.mdl"

```

```

strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objLocationsDim = objModel.Dimensions("Sales regions")
'Remove the Branch Code level and then use it in an alternate
drill-down path.
objLocationsDim.DrillDowns(1).Levels("Branch").Delete()
'Obtain the convergence level and create the alternate drill-down
path.
objLevel = objLocationsDim.DrillDowns(1).Levels("Employee")
With objLevel
    .Unique = True
    .UniqueMove = True
    .Update()
End With
objAltDrill = objLevel.CreateAlternateDrillDown
'Add a new level to the Levels collection of the alternate drill-down
path.
objNewLevel = objAltDrill.Levels.Add(xtrObjectType.trLevel)
'Associate the new level with columns in the data source.
objRefSource = objModel.DataSources("Sales region (csv)").Columns("Branch code")
objRefLabel = objModel.DataSources("Sales region (csv)").Columns("Branch city")
With objNewLevel
    .AssociateWith(objRefSource, xtrAssociationRole.trAssociationSource)
    .AssociateWith(objRefLabel, xtrAssociationRole.trAssociationLabel)
    .BlankSubstitute = "No Value"
    .Inclusion = xtrInclusion.trInclusionGenerate
    .NewCatsLocked = False
    .Name = "Branch"
    .RefreshDescription = True
    .RefreshLabel = True
    .RefreshShortName = True
    .Update()
End With
'Move the level and then set the primary drill down.
With objAltDrill
    .Levels("Branch").Move(1)
    .IsPrimary = True
    .Name = "By Branch"
    .Update()
End With
With objModel
    .GenerateCategories()
    .Update()
    .SaveAs("Great outdoors 8X.mdl")
    .Close()
End With
objRefLabel = Nothing
objRefSource = Nothing
objNewLevel = Nothing
objAltDrill = Nothing
objLevel = Nothing
objLocationsDim = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple d'utilisation de l'objet DateWizard pour créer une dimension de temps

L'exemple suivant crée un modèle, ajoute une source de données et utilise l'objet DateWizard pour créer une dimension de temps.

```
Sub Example21()  
    Dim objTransApp As Object  
    Dim objModel As Object  
    Dim objDataSource As Object  
    Dim objColumn As Object  
    Dim objDateWizard As Object  
    Dim objDateDim As Object  
    Dim objDrillDown As Object  
    Dim strIBMCognos10Location As String  
    Dim strDataSource As String  
    Dim strDataPath As String  
    'Change these paths to match your installation  
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\  
    strDataSource = "NATIONAL.ASC"  
    strDataPath = strIBMCognos10Location &  
        "webcontent\samples\datasources\cubes\PowerCubes\EN\National\  
&  
        strDataSource  
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")  
    objModel = objTransApp.NewModel  
    objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)  
    With objDataSource  
        .DecimalPoint = "."  
        .LocalPath = strDataPath  
        .Name = "National"  
        .SourceType = xtrSourceType.trFlatFileColumnNames  
        .Update()  
    End With  
    objColumn = objDataSource.Columns.Item("Date")  
    objDateWizard = objModel.DateWizard  
    With objDateWizard  
        .DimensionName = "Dates"  
        .DimensionAssociateWith(objColumn, xtrAssociationRole.trAssociationSource)  
        .EarliestDate = 19930101  
        .EnableTimePeriod(xtrTimeType.trTimeTypeYear) = True  
        .EnableTimePeriod(xtrTimeType.trTimeTypeQuarter) = True  
        .EnableTimePeriod(xtrTimeType.trTimeTypeMonth) = True  
        .EnableTimePeriod(xtrTimeType.trTimeTypeWeek) = False  
        .EnableTimePeriod(xtrTimeType.trTimeTypeDay) = False  
        .GenerateDates = False  
        .LatestDate = 19941231  
        .MonthType = xtrSpecialFunction.trSpecialFunctionMonth  
        .QuarterType = xtrSpecialFunction.trSpecialFunctionQuarter  
        .WeekAdd = xtrWeekAdd.trWeekAddNone  
        .WeekSpan = xtrWeekSpan.trWeekSpanNone  
        .WeekStartDay = xtrWeekDay.trMonday  
        .WorkingDays(127)  
        .WorkingDay(xtrWeekDay.trSunday) = False  
        .YearType = xtrSpecialFunction.trSpecialFunctionYear  
    End With  
    objDateDim = objDateWizard.CreateDateDimension()  
    objDateDim.Update()  
    objDrillDown = objDateDim.DrillDowns(1)  
    With objDrillDown  
        .DrillCode = "By Dates"  
        .DrillInclusion = xtrInclusion.trInclusionSuppress  
        .WeekSpan = xtrWeekSpan.trWeekSpanSplitMost  
        .WeekStart = xtrWeekDay.trMonday  
        .Update()  
    End With
```

```

With objModel
    .GenerateCategories()
    .SaveAs("NationalX.mdl")
    MsgBox(.FileName & " " & .Size & " " & .Time)
    .Close()
End With
objDrillDown = Nothing
objDateDim = Nothing
objDateWizard = Nothing
objColumn = Nothing
objDataSource = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une mesure de nombre de catégories et d'ajout à un modèle

L'exemple suivant crée une mesure de nombre de catégories et l'ajoute à un modèle existant.

```

Sub Example22()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objMeasure As Object
    Dim objLevel As Object
    Dim objActMeasure As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String

    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    objMeasure = objModel.Measures.Add()
    objLevel = objModel.Dimensions("Sales region").DrillDowns(1).Levels("Branch")
    With objLevel
        .Unique = True
        .Update()
    End With
    objActMeasure = objModel.Measures("Quantity")
    objMeasure.ActivityMeasure = objActMeasure
    objMeasure.CategoryCountLevel = objLevel
    With objMeasure
        .AllowCurrencyConversion = False
        .AllowDrillThrough = False
        .Name = "Sales branch count"
        .OutputScale = 0
        .Precision = 0
        .ReverseSign = False
        .StorageType = xtrStorage.trStorageDefault
        .Update()
    End With
    objModel.SaveAs("great outdoors salesX.mdl")
    objModel.Close()
    objActMeasure = Nothing
    objLevel = Nothing
End Sub

```

```

objMeasure = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout d'une vue des dimensions à un modèle

L'exemple suivant ajoute une vue des dimensions à un modèle.

```

Sub Example23()
Dim objTransApp As Object
Dim objModel As Object
Dim objProductsDim As Object
Dim objViewItem As Object
Dim objCube As Object
Dim strIBMCognos10Location As String
Dim strModelPath As String
Dim strModelSource As String
'Change these paths to match your installation
strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
strModelSource = "great outdoors sales.mdl"
strModelPath = strIBMCognos10Location & _
    "webcontent\samples\models\Transformer8\EN\" & strModelSource
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objProductsDim = objModel.Dimensions.Item("Products")
objViewItem = objProductsDim.Views.Add()
objViewItem.Apex = objProductsDim.Drilldowns(1).Categories(1)
With objViewItem
    .Name = "OutDoor Products"
    .ViewType = xtrViewType.trViewTypeCustom
    .Update()
End With
objProductsDim.Update()
objCube = objModel.Cubes(1)
objCube.DimensionView(objProductsDim) = objViewItem
objCube.Update()
With objModel
    .SaveAs("great outdoors salesX.mdl")
    .Close()
End With
objCube = Nothing
objViewItem = Nothing
objProductsDim = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de déplacement d'une catégorie enfant vers un autre parent

L'exemple suivant transfère une catégorie enfant vers un autre parent pour réaffecter un pays ou une région à une autre région.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie "objCategory =", vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte". Veuillez utiliser plutôt l'interface utilisateur.



```

Sub Example24()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objCategory As Object
    Dim objCatToMove As Object
    Dim objCatToReceive As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Sales and Marketing.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    objModel = objTransApp.OpenModel(strModelPath)
    'Get the Asia Pacific region category.
    objCategory = objModel.Dimensions("Retailers").DrillDowns(1).Categories(2)
    'Get the Australia category.
    objCatToMove = objCategory.ChildCategories(5)
    'Get the Americas region category.
    objCatToReceive = objModel.Dimensions("Retailers").DrillDowns(1).Categories(1)
    'Move Australia to a new region.
    objCatToMove.MoveToCategory(objCatToReceive)
    objModel.SaveAs("Sales and MarketingX.mdl")
    objModel.Close()
    objCatToReceive = Nothing
    objCatToMove = Nothing
    objCategory = Nothing
    objModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout d'une table dans un fichier et de chargement des données

L'exemple suivant crée un petit fichier d'exemple nommé EurosX.csv, qui contient les taux de conversion de plusieurs devises de l'UEM (Union économique et monétaire). Il contient aussi les taux de base pour la conversion entre la devise de base et l'euro, puis ajoute une table en euro et utilise EurosX.csv pour charger la table avec les données.

```

Sub Example25()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDataSource As Object
    Dim objEuroTable As Object
    Dim objColumn As Object
    Dim objCurrencyRec As Object
    Dim intX As Integer
    Dim FileNumber As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    Dim strDataPath As String
    Dim strDataSource As String
    Dim strDataSourceName As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "great outdoors sales.mdl"
    strModelPath = strIBMCognos10Location & _

```

```

        "webcontent\samples\models\Transformer8\EN\" & strModelSource
strDataSource = "Euro CurrencyX.csv"
strDataPath = System.IO.Directory.GetCurrentDirectory() &

-
    "\\.\..\..\\" & strDataSource
strDataSourceName = "Euro Currency"
'Create Eurofile
FileNumber = FreeFile()
FileOpen(FileNumber, strDataPath, OpenMode.Output)
PrintLine(FileNumber, "EuroDate,EuroCurrency,EuroCode,EuroConRate")
PrintLine(FileNumber, "0,Austrian Schilling,AUT,13.7603")
PrintLine(FileNumber, "0,French Franc,FRA,6.55957")
PrintLine(FileNumber, "20040101,U.S. Dollar,USA,1.2597")
PrintLine(FileNumber, "20040201,U.S. Dollar,USA,1.2452")
FileClose(FileNumber)
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
objDataSource = objModel.DataSources.Add(xtrObjectType.trFlatFileDataSource)
With objDataSource
    .GenerateCategories = False
    .GeneratePowerCube = xtrPowerCubeGeneration.trGenerationNoCreatePowerCubes
    .LocalPath = strDataPath
    .Name = strDataSourceName
    .SourceType = xtrSourceType.trFlatFileColumnNames
    .SetsCurrentPeriod = False
    .Update()
    .Columns("EuroDate").DataClass = xtrDataClass.trDataClassDate
    .Columns("EuroDate").DateDegreeofDetail = xtrDateLevel.trDateLevelMonth
    .Columns("EuroDate").DateInputFormat = xtrDateFormat.trMDY
    .Columns("EuroDate").Update()
End With
objEuroTable = objModel.CurrencyTables.Add()
With objEuroTable
    .CurrencyTableType = xtrCurrencyTableType.trCurrencyTableEuro
    .Update()
End With
objColumn = _
objModel.DataSources(strDataSourceName).Columns("EuroCurrency")
objEuroTable.AssociateWith(objColumn, xtrAssociationRole.trAssociationLabel)
objColumn = _
objModel.DataSources(strDataSourceName).Columns("EuroConRate")
objEuroTable.AssociateWith(objColumn, xtrAssociationRole.trAssociationRate)
objColumn = _
objModel.DataSources(strDataSourceName).Columns("EuroCode")
objEuroTable.AssociateWith(objColumn,
xtrAssociationRole.trAssociationCountryCode)
objColumn = _
objModel.DataSources(strDataSourceName).Columns("EuroDate")
objEuroTable.AssociateWith(objColumn,
xtrAssociationRole.trAssociationCurrencyDate)
objEuroTable.Update()
objModel.LoadCurrencyTable()
For intX = 1 To objModel.CurrencyRecords.Count
    objCurrencyRec = objModel.CurrencyRecords(intX)
    Select Case objCurrencyRec.CountryCode
        Case "AUT", "FIN", "FRA"
            With objCurrencyRec
                .CurrencyIsEMU = True
                .EMUEntryDate = "19990101"
                .Update()
            End With
        Case Else
    ,

```

```

        End Select
    Next intX
    With objModel
        .LoadCurrencyTable()
        .SaveAs("great outdoors salesX.mdl")
        .Close()
    End With
    objCurrencyRec = Nothing
    objColumn = Nothing
    objEuroTable = Nothing
    objDataSource = Nothing
    objModel = Nothing
    objTransApp = Nothing
End Sub

```

---

## Exemple de génération d'une dimension de temps basée sur une année lunaire

L'exemple suivant crée un modèle, ajoute une source de données et génère une dimension de temps basée sur une année lunaire.

```

Sub Example26()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDataSource As Object
    Dim objColumn As Object
    Dim objDateWizard As Object
    Dim objDateDim As Object
    Dim objDrillDown As Object
    Dim objDateLevel As Object
    Dim strIBMCognos10Location As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strDataSource = "NATIONAL.ASC"
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\National\"
    & _
        strDataSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    objModel = objTransApp.NewModel
    objDataSource = objModel.DataSources.Add()
    With objDataSource
        .LocalPath = strDataPath
        .Name = "National"
        .SourceType = xtrSourceType.trFlatFileColumnNames
        .Update()
    End With
    objColumn = objDataSource.Columns.Item("Date")
    objDateWizard = objModel.DateWizard
    With objDateWizard
        .DimensionName = "Dates"
        .DimensionAssociateWith(objColumn, xtrAssociationRole.trAssociationSource)
        .EarliestDate = 19900101
        .EnableTimePeriod(xtrTimeType.trTimeTypeMonth) = True
        .EnableTimePeriod(xtrTimeType.trTimeTypeQuarter) = True
        .EnableTimePeriod(xtrTimeType.trTimeTypeYear) = True
        .LatestDate = 20101231
        .WorkingDays = xtrWeekDay.trMonday + xtrWeekDay.trTuesday
    + _
        xtrWeekDay.trWednesday + xtrWeekDay.trThursday
    + _
        xtrWeekDay.trFriday
    End With
    objDateDim = objDateWizard.CreateDateDimension()

```

```

objDateDim.Update()
objDrillDown = objDateDim.DrillDowns(1)
With objDrillDown
    .DrillCode = "By Order Date"
    .DrillInclusion = xtrInclusion.trInclusionSuppress
    .Lunar = True
    .WeekAdd = xtrWeekAdd.trWeekAddDays4
    .WeekStart = xtrWeekDay.trMonday
    .YearStartDay = 19900101
    .Update()
End With
objDateLevel = objDateDim.DimensionLevels(1)
objDateLevel.DateFunction = xtrSpecialFunction.trSpecialFunctionLunarYear
objDateLevel.Update()
objDateLevel = objModel.Dimensions("Dates").DimensionLevels(2)
objDateLevel.DateFunction = xtrSpecialFunction.trSpecialFunctionLunarQuarter
objDateLevel.Update()

objDateLevel = objModel.Dimensions("Dates").DimensionLevels(3)
objDateLevel.DateFunction = xtrSpecialFunction.trSpecialFunctionLunarMonth445
objDateLevel.Update()
With objModel
    .GenerateCategories()
    .SaveAs("NationalX.mdl")
    .Close()
End With
objDateLevel = Nothing
objDrillDown = Nothing
objDateDim = Nothing
objDateWizard = Nothing
objColumn = Nothing
objDataSource = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de déplacement d'une catégorie enfant vers une autre catégorie parent

L'exemple suivant déplace une catégorie enfant vers une nouvelle catégorie parent dans le même niveau.

**Remarque :** dans certains cas, vous ne pouvez pas utiliser de macro pour définir les propriétés d'un objet de catégorie. La macro peut paraître correcte mais lorsque vous exécutez la partie "objCategory =", vous recevez le message suivant : "Cette collection est en lecture seule dans ce contexte". Veuillez utiliser plutôt l'interface utilisateur.

```

Sub Example27()
    Dim objTransApp As Object
    Dim objModel As Object
    Dim objDimension As Object
    Dim objCategories As Object
    Dim objCategory As Object
    Dim objChildCategory As Object
    Dim objParentCategory As Object
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Sales and Marketing.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")

```

```

With objTransApp
    .DataSourcePath = strIBMCognos10Location & "bin"
    .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
End With
objModel = objTransApp.OpenModel(strModelPath)
'Get a dimension
objDimension = objModel.Dimensions("Retailers")
'Get a collection of categories
objCategories = objDimension.DrillDowns(1).Categories
'Get the 5th category
objCategory = objCategories(5)
'Find the Spain child category
For child_category_index = 1 To objCategory.ChildCategories.Count
    objChildCategory = objCategory.ChildCategories(child_category_index)
    If objChildCategory.Name = "Spain" Then
        'Find the Americas category and move the Spain child
category to it
        For parent_category_index = 1 To objCategories.Count
            objParentCategory = objCategories(parent_category_index)
            If objParentCategory.Name = "Americas" Then
                objChildCategory.ConnectWithCategory(objParentCategory)
                Exit For
            End If
        Next
    Exit For
End If
Next
With objModel
    .SaveAs("Sales and MarketingX.mdl")
    .Close()
End With
objParentCategory = Nothing
objChildCategory = Nothing
objCategory = Nothing
objCategories = Nothing
objDimension = Nothing
objModel = Nothing
objTransApp = Nothing
End Sub

```

---

## Exemple de définition d'attributs pour une application

L'exemple suivant montre comment définir plusieurs attributs de l'objet Application.

```

Sub Example28()
Dim objTransApp As Object
Dim strStartLocation As String
strStartLocation = System.IO.Directory.GetCurrentDirectory()
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
With objTransApp
    .DataCharacterSet = xtrCharacterType.trCharDefault
    .DataSourcePath = strStartLocation
    .DataTemporaryFilesPath = strStartLocation
    .DefaultDateFormat = xtrPreferences.trDateFormatFromControlPanel
    .DetachDataSource = True
    .EnableMessageLogging = True
    .LogErrorLevel = xtrPreferences.trLogErrorsAndAbove
    .LogFileAppend = True
    .LogFileName = "TrModelsLog.log"
    .LogFilesPath = strStartLocation
    .MaxTransactionNumber = 500000
    .ModelsPath = strStartLocation
    .ModelTemporaryFilesPath = strStartLocation
    .PowerCubesPath = strStartLocation
    .RowsAsSample = 600

```

```

        .RowsChecked = 600
        .SortComparisonRule = xtrPreferences.trSortIgnoreControlPanel
    End With
    objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout et de suppression d'un pack

L'exemple suivant ouvre un modèle existant, y ajoute deux packs, affiche les informations relatives à chacun de ces packs, supprime l'un des packs, puis enregistre le modèle dans un nouveau fichier.

```

Sub Example29()
    Dim objTransApp As Object
    Dim model As Model
    Dim packages As Packages
    Dim package As Package
    Dim new_package As Package
    Dim temp_package As Package
    Dim path As String
    Dim timestamp As String
    Dim name As String
    Dim index As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Sales and Marketing.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    'Open a package based model
    model = objTransApp.OpenModel(strModelPath)
    'Create a new package and add it to the model
    timestamp = Format(Now, "yyyy-MM-ddTHH:mm:ss.fffZ")
    new_package = model.Packages.Add()
    new_package.Name = "GO Data Warehouse (analysis)"
    new_package.Path = "/content/package[@name='GO Data Warehouse (analysis)']"
    new_package.TimeStamp = "/content/package[@name='GO Data Warehouse (analysis)']"
    -
        & "/model[@name='" & timestamp & "']"
    new_package.Update()
    'Create a second package and add it to the model
    timestamp = Format(Now, "yyyy-MM-ddTHH:mm:ss.fffZ")
    temp_package = model.Packages.Add()
    temp_package.Name = "Temporary Package"
    temp_package.Path = "/content/package[@name='Temporary Package']"
    temp_package.TimeStamp = "/content/package[@name='Temporary Package']"
    Package']"
        & "/model[@name='" & timestamp & "']"
    temp_package.Update()
    'Iterate through all packages and display information
    packages = model.Packages
    For index = 1 To packages.Count
        package = packages.Item(index)
        name = package.Name
        path = package.Path
        timestamp = package.TimeStamp
        MsgBox("Package name: " & name & Chr(13) & _
            "Package path: " & path & Chr(13) & _

```

```

        "Package time stamp: " & timestamp)
Next
'Delete second package
packages.Remove(temp_package)
model.Update()
'Save the model under a different location
model.SaveAs("Sales and MarketingX.mdl")
model.Close()
objTransApp = Nothing
End Sub

```

---

## Exemple d'ajout et de suppression d'un rapport

L'exemple suivant ouvre un modèle existant, y ajoute deux rapports, affiche les informations relatives à chacun de ces rapports, supprime l'un des rapports, puis enregistre le modèle dans un nouveau fichier.

```

Sub Example30()
    Dim objTransApp As Object
    Dim model As Model
    Dim reports As Reports
    Dim report As Report
    Dim new_report As Report
    Dim temp_report As Report
    Dim path As String
    Dim timestamp As String
    Dim name As String
    Dim index As Integer
    Dim strIBMCognos10Location As String
    Dim strModelPath As String
    Dim strModelSource As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strModelSource = "Employee expenses.mdl"
    strModelPath = strIBMCognos10Location & _
        "webcontent\samples\models\Transformer8\EN\" & strModelSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    'Open a report based model
    model = objTransApp.OpenModel(strModelPath)
    'Create a new report and add it to the model
    timestamp = Format(Now, "yyyy-MM-ddTHH:mm:ss.fffZ")
    new_report = model.Reports.Add()
    new_report.Name = "TOC report"
    new_report.Path = "/content/package[@name='GO Data Warehouse
(query)']" & _
        & "/folder[@name='Report Studio Report Samples']" & _
        & "/report[@name='TOC report']"
    new_report.TimeStamp = timestamp
    new_report.Update()
    'Create a second report and add it to the model
    timestamp = Format(Now, "yyyy-MM-ddTHH:mm:ss.fffZ")
    temp_report = model.Reports.Add()
    temp_report.Name = "Temporary Report"
    temp_report.Path = "/content/package[@name='Temporary Package']"
    -
        & "/report[@name='Temporary Report']"
    temp_report.TimeStamp = timestamp
    temp_report.Update()
    'Iterate through all reports and display information

```

```

reports = model.Reports
For index = 1 To reports.Count
    report = reports.Item(index)
    name = report.Name

    path = report.Path
    timestamp = report.TimeStamp
    MsgBox("Report name: " & name & Chr(13) & _
        "Report path: " & path & Chr(13) & _
        "Report time stamp: " & timestamp)
Next
'Delete second package
reports.Remove(temp_report)
model.Update()
'Save the model under a different location
model.SaveAs("Employee expensesX.mdl")
model.Close()
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une requête

L'exemple suivant montre comment créer une requête dans le cadre de la création d'un petit modèle.

```

Sub Example31()
    Dim objTransApp As Object
    Dim model As Model
    Dim new_package As Package
    Dim new_query As Query
    Dim new_column1 As Column
    Dim new_column2 As Column
    Dim new_column3 As Column
    Dim measure As Measure
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    'Create a new model
    model = objTransApp.NewModel
    'Create a new package and add it to the model
    new_package = model.Packages.Add()
    With new_package
        .Name = "GO Data Warehouse (query)"
        .Path = "/content/package[@name='GO Data Warehouse (query)']"
        .Update()
    End With
    'Create a new query and add it to the package
    new_query = new_package.Queries.Add()
    With new_query
        .Name = "Sales Revenue"
        .AutoSummary = True
        .MaximizeSpeed = True
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Create 3 new columns and add them to the query
    new_column1 = new_query.Columns.Add()
    With new_column1
        .Name = "Date"
        .OriginalName = "[Sales (query)].[Time dimension].[Date]"
        .Update()
    End With
    new_column2 = new_query.Columns.Add()
    With new_column2
        .Name = "Region"
        .OriginalName = "[Sales (query)].[Retailer site].[Region]"
        .Update()
    End With
    new_column3 = new_query.Columns.Add()

```



```

With new_column3
    .Name = "Revenue"
    .OriginalName = "[Sales (query)].[Sales fact].[Revenue]"
    .Update()
End With
'Add Revenue column as a measure
measure = model.Measures.Add()
With measure
    .AssociateWith(new_column3, xtrAssociationRole.trAssociationSource)
    .Update()
End With
model.DoAutoDesign()
' See if there is any columns that are mismatched
If new_query.IsAnyColumnMismatched = True Then
    MsgBox("There is at least one column that is mismatched.")
    For index = 1 To model.CheckModel.Count
        MsgBox(model.CheckModel.Item(index).Name)
    Next index
End If
'Save the model
With model
    .SaveAs("GO Data Warehouse (query).mdl")
    .Close()
End With
objTransApp = Nothing
End Sub

```

---

## Exemple de création et de suppression des filtres d'un modèle

L'exemple suivant montre comment créer et supprimer des filtres dans le cadre de la création d'un petit modèle.

```

Sub Example32()
    Dim objTransApp As Object
    Dim model As Model
    Dim new_package As Package
    Dim new_query As Query
    Dim new_column As Column
    Dim new_filter As Filter
    Dim filters As Filters
    Dim current_filter As Filter
    Dim measure As Measure
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    'Create a new model
    model = objTransApp.NewModel
    'Create a new package and add it to the model
    new_package = model.Packages.Add()
    With new_package
        .Name = "GO Data Warehouse (query)"
        .Path = "/content/package[@name='GO Data Warehouse (query)']"
        .Update()
    End With
    'Create a new query and add it to the package
    new_query = new_package.Queries.Add()
    With new_query
        .Name = "Sales Revenue"
        .AutoSummary = True
        .MaximizeSpeed = True
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Create 3 new columns and add them to the query
    new_column = new_query.Columns.Add()
    With new_column
        .Name = "Date"
        .OriginalName = "[Sales (query)].[Time dimension].[Date]"
        .Update()
    End With

```

```

End With
new_column = new_query.Columns.Add()
With new_column
    .Name = "Region"
    .OriginalName = "[Sales (query)].[Retailer site].[Region]"
    .Update()
End With
new_column = new_query.Columns.Add()
With new_column
    .Name = "Revenue"
    .OriginalName = "[Sales (query)].[Sales fact].[Revenue]"
    .Update()
End With
'Create 3 new filters
new_filter = new_query.Filters.Add()
With new_filter
    .Name = "2004"
    .RefName = "[go_data_warehouse].[2004]"
    .Update()
End With
new_filter = new_query.Filters.Add()
With new_filter
    .Name = "2005"
    .RefName = "[go_data_warehouse].[2005]"
    .Update()
End With
new_filter = new_query.Filters.Add()
With new_filter
    .Name = "GO Asia Pacific"
    .RefName = "[go_data_warehouse].[GO Asia Pacific]"
    .Update()
End With
'Add Revenue column as a measure
measure = model.Measures.Add()
With measure
    .AssociateWith(new_column, xtrAssociationRole.trAssociationSource)
    .Update()
End With
>Delete the 2005 filter
filters = new_query.Filters
For index = 1 To filters.Count
    current_filter = filters.Item(index)
    If current_filter.Name = "2005" Then
        current_filter.Delete()
    Exit For
End If
Next
model.DoAutoDesign()
'Save the model
With model
    .SaveAs("GO Data Warehouse (query).mdl")
    .Close()
End With
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une invite à valeur unique

L'exemple suivant montre comment créer des invites à valeur unique dans le cadre de la création d'un petit modèle.

```

Sub Example33()
    Dim objTransApp As Object
    Dim model As Model
    Dim new_report As Report
    Dim new_query As Query

```

```

Dim new_column As Column
Dim new_prompt As Prompt
objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
'Create a new model
model = objTransApp.NewModel
'Create a new report and add it to the model
new_report = model.Reports.Add()
new_report.Name = "Date Between List"
new_report.Path = "/content/package[@name='GO Data Warehouse
(query)']" & "/folder[@name='SDK Report Samples']" & _
& "/report[@name='Date Between List']"
new_report.Update()
'Create a new query and add it to the report
new_query = new_report.Queries.Add()
With new_query
.Name = "Date Between List (Has Prompt)"
.AutoSummary = False
.MaximizeSpeed = True
.SetsCurrentPeriod = True
.Update()
End With
'Create 3 new columns and add them to the query
new_column = new_query.Columns.Add()
With new_column
.Name = "Date"
.OriginalName = "[Report].[Query1.0].[Date]"
.Update()
End With
new_column = new_query.Columns.Add()
With new_column
.Name = "Order number"
.OriginalName = "[Report].[Query1.0].[Order number]"
.Update()
End With
new_column = new_query.Columns.Add()
With new_column
.Name = "Order method"
.OriginalName = "[Report].[Query1.0].[Order method]"
.Update()

End With
'Create 2 new Single-valued prompts and add them to the query
new_prompt = new_query.Prompts.Add()
With new_prompt
.PromptValueType = xtrPromptValueType.trSingleValuePrompt
.Name = "StartDate"
.Type = "Date Time (1999-01-31 18:30:00)"
.Value = "2005-01-01"
.Update()
End With
new_prompt = new_query.Prompts.Add()
With new_prompt
.PromptValueType = xtrPromptValueType.trSingleValuePrompt
.Name = "EndDate"
.Type = "Date Time (1999-01-31 18:30:00)"
.Value = "2006-12-31"
.Update()
End With
'Save the model
With model
.SaveAs("GO Data Warehouse (query)X.mdl")
.Close()
End With
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'une invite à valeurs multiples

L'exemple suivant montre comment créer une invite à valeurs multiples dans le cadre de la création d'un petit modèle.

```
Sub Example34()
    Dim objTransApp As Object
    Dim model As Model

    Dim new_report As Report
    Dim new_query As Query
    Dim new_column As Column
    Dim new_prompt As Prompt
    Dim list_of_values As String = ""
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    'Create a new model
    model = objTransApp.NewModel
    'Create a new report and add it to the model
    new_report = model.Reports.Add()
    new_report.Name = "Product Method Date List"
    new_report.Path = "/content/package[@name='GO Data Warehouse
(query)']"
        & "/folder[@name='SDK Report Samples']"
        & "/report[@name='Product Method Date List']"
    new_report.Update()
    'Create a new query and add it to the report
    new_query = new_report.Queries.Add()
    With new_query
        .Name = "Product Method Date List (Has Prompt)"
        .AutoSummary = False
        .MaximizeSpeed = True
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Create 3 new columns and add them to the query
    new_column = new_query.Columns.Add()
    With new_column
        .Name = "Date"
        .OriginalName = "[Report].[Query1.0].[Date]"
        .Update()
    End With
    new_column = new_query.Columns.Add()
    With new_column
        .Name = "Product Name"
        .OriginalName = "[Report].[Query1.0].[Product name]"
        .Update()
    End With
    new_column = new_query.Columns.Add()
    With new_column
        .Name = "Order method"
        .OriginalName = "[Report].[Query1.0].[Order method]"
        .Update()
    End With
    'Create a new Multi-valued prompt and add it to the query
    new_prompt = new_query.Prompts.Add()
    With new_prompt
        .PromptValueType = xtrPromptValueType.trMultiValuePrompt
        .Name = "MP"
        .Type = "String"
        .Value = "Fax"
        .Value = "Mail"
        .Value = "Telephone"
        .Value = "Web"
        .Update()
    End With
    'List all the prompt values
    For index = 1 To new_prompt.ValuesCount
```

```

        new_prompt.CurrentValueIndex = index
    If list_of_values <> "" Then
        list_of_values = list_of_values & ", " & new_prompt.Value
    Else
        list_of_values = new_prompt.Value
    End If
Next
MsgBox("These are the prompt values: " & list_of_values)
'Save the model
With model
    .SaveAs("GO Data Warehouse (query).mdl")
    .Close()
End With
objTransApp = Nothing
End Sub

```

---

## Exemple de création d'un modèle et de publication d'un PowerCube

L'exemple suivant utilise la source de données NATIONAL.ASC pour créer un modèle et publier un PowerCube sur le serveur.

```

Sub Example35()
    Dim objTransApp As Object
    Dim model As Model
    Dim cube As Cube
    Dim dataSource As FlatFileDataSource
    Dim strIBMCognos10Location As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strDataSource = "NATIONAL.ASC"
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\National\"
    & _
        strDataSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    objTransApp.Logon("Cognos", "", "")
    'Create new model and add flat file data source
    model = objTransApp.NewModel
    dataSource = model.DataSources.Add(xtrObjectType.trFlatFileDataSource)
    With dataSource
        .GenerateCategories = True
        .GeneratePowerCube = xtrPowerCubeGeneration.trGenerationDefault
        .CharacterType = xtrCharacterType.trCharAnsiSingleByte
        .External = False
        .FieldSeparator = ","
        .LocalPath = strDataPath
        .MaximizeSpeed = True
        .Name = "National"
        .SourceType = xtrSourceType.trFlatFileColumnNames
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Do Autodesign and make National cube
    With model
        .Name = "National"
        .DoAutoDesign()
        .TestBuild(20, True)
        .Update()
    End With
    'Publish the National Cube
    cube = model.Cubes.Item("National")
    With cube
        .MDCFile = "c:\National.mdc"
        .Update()
    End With
End Sub

```

```

        .CreateMDCFile()
        .PublishPackage(True, True) 'Re-publish both Datasource
& Package
    End With
    'Save model
    With model
        .SaveAs("NationalX.mdl")
        .Close()
    End With
    objTransApp.Logoff()
    objTransApp = Nothing
End Sub

```

---

## Exemple de copie et d'activation d'un PowerCube

L'exemple suivant montre comment utiliser la fonction de copie et d'activation.

```

Sub Example36()
    Dim objTransApp As Object
    Dim model As Model
    Dim cube As Cube
    Dim dataSource As FlatFileDataSource
    Dim strIBMCognos10Location As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strDataSource = "NATIONAL.ASC"
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\National\"
    & _
        strDataSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    objTransApp.Logon("Cognos", "", "")
    'Create new model and add flat file data source
    model = objTransApp.NewModel
    dataSource = model.DataSources.Add(xtrObjectType.trFlatFileDataSource)
    With dataSource
        .GenerateCategories = True
        .GeneratePowerCube = xtrPowerCubeGeneration.trGenerationDefault
        .CharacterType = xtrCharacterType.trCharAnsiSingleByte
        .External = False
        .FieldSeparator = ","
        .LocalPath = strDataPath
        .MaximizeSpeed = True
        .Name = "National"
        .SourceType = xtrSourceType.trFlatFileColumnNames
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Do Autodesign and make National cube
    With model
        .Name = "National"
        .DoAutoDesign()
        .TestBuild(20, True)
        .Update()
    End With
    'Copy on Activate the National cube
    cube = model.Cubes.Item("National")
    With cube
        .MDCFile = "c:\National.mdc"
        .Update()
        .CreateMDCFile()
        .SetDeployType(xtrDeployType.trDeployType_SWAPSINGLE)
        .AddDeployLocation("c:\NATIONAL\Deployment1")
        .AddDeployLocation("c:\NATIONAL\Deployment2")
    End With
End Sub

```

```

        .AddDeployLocation("c:\NATIONAL\Deployment3")
        .DataSourceWindowsLocation = "c:\NATIONAL\Deployment1\National.mdc"
        .Update()
        .DeployCube()
        .PublishDatasource(True) 'Republish PowerCube if it already
exists
        .ClearDeployLocations() 'Remove the deployment locations
from the model
        .Update()
        End With
        'Save model
        With model
            .SaveAs("NationalX.mdl")
            .Close()
        End With
        objTransApp.Logoff()
        objTransApp = Nothing
    End Sub

```

---

## Exemple de création d'un modèle avec un code d'accès et une source de données IQD

L'exemple suivant montre comment créer un modèle avec un code d'accès et une source de données IQD.

```

Sub Example37()
    Dim objTransApp As Object
    Dim model As Model
    Dim datasource As IqdDataSource
    Dim signon As Signon
    Dim strIBMCognos10Location As String
    Dim strDataSource As String
    Dim strDataPath As String
    'Change these paths to match your installation
    strIBMCognos10Location = "C:\Program Files\IBM\Cognos\c10\"
    strDataSource = "prod.iqd"
    strDataPath = strIBMCognos10Location & _
        "webcontent\samples\datasources\cubes\PowerCubes\EN\great
outdoors sales\" & _
        strDataSource
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    With objTransApp
        .DataSourcePath = strIBMCognos10Location & "bin"
        .TransdaPath = strIBMCognos10Location & "CS7Gateways\bin\TransDa.exe"
    End With
    model = objTransApp.NewModel
    'Create signon before adding the data source
    signon = model.Signons.Add()
    With signon
        .Name = "great_outdoors_warehouse"
        .Description = "Signon used for prod.iqd data source"
        .UserID = "sa"
        .Password = "sa"
        .PromptForPassword = False
        .SignonType = xtrSignonType.trDataSourceSignon
        .Update()
    End With
    'Add IQD data source
    datasource = model.DataSources.Add(xtrObjectType.trIqdDataSource)
    With datasource
        .GenerateCategories = True
        .GeneratePowerCube = xtrPowerCubeGeneration.trGenerationDefault
        .IsolationLevel = 0
        .LocalPath = strDataPath
        .Name = "Products"
        .SetsCurrentPeriod = True
    End With

```

```

        .SourceType = xtrSourceType.trQuery
        .Update()
    End With
    'Auto Design, build cube, and save model
    With model
        .Name = "Products"
        .DoAutoDesign()
        .TestBuild(20, True)
        .Update()
        .SaveAs("ProductsX.mdl")
        .Close()
    End With
    objTransApp = Nothing
End Sub

```

---

## Exemple de création d'un modèle avec un code d'accès et une source de données de pack

L'exemple suivant montre comment créer un modèle avec un code d'accès et une source de données de pack.

```

Sub Example38()
    Dim objTransApp As Object
    Dim model As Model
    Dim package As Package
    Dim query As Query
    Dim column As Column
    Dim signon As Signon
    Dim connection As PackageDatasourceConnection
    Dim measure As Measure
    objTransApp = CreateObject("IBMCognosTransformer.ApplicationCtrl.1")
    'Create a new model
    model = objTransApp.NewModel
    'Create a new package and add it to the model
    package = model.Packages.Add()
    With package
        .Name = "G0 Data Warehouse (query)"
        .Path = "/content/package[@name='G0 Data Warehouse (query)']"
        .Update()
    End With

    'Create signon before adding the PackageDatasourceConnection
    signon = model.Signons.Add()
    With signon
        .Name = "great_outdoors_warehouse"
        .Description = "Signon used for package data source"
        .SignOnNamespace = "Cognos"
        .UserID = "sa"
        .Password = "sa"
        .SignonType = xtrSignonType.trCognosSignon
        .AutoLogon = True
        .Update()
    End With

    'Add PackageDatasourceConnection
    connection = package.PackageDatasourceConnections.Add()
    With connection
        .DataSource = "great_outdoors_warehouse"
        .Connection = "great_outdoors_warehouse"
        .Signon = "great_outdoors_warehouse"
        .TransformerSignon = "great_outdoors_warehouse"
        .AlwaysUseTransformerSignon = True
        .Update()
    End With

    'Create a new query and add it to the package
    query = package.Queries.Add()
    With query
        .Name = "Sales Revenue"
    End With
End Sub

```



```

        .AutoSummary = True
        .MaximizeSpeed = True
        .SetsCurrentPeriod = True
        .Update()
    End With
    'Create 3 new columns and add them to the query
    column = query.Columns.Add()
    With column
        .Name = "Date"
        .OriginalName = "[Sales (query)].[Time dimension].[Date]"
        .Update()
    End With
    column = query.Columns.Add()
    With column
        .Name = "Region"
        .OriginalName = "[Sales (query)].[Retailer site].[Region]"
        .Update()
    End With
    column = query.Columns.Add()
    With column
        .Name = "Revenue"
        .OriginalName = "[Sales (query)].[Sales fact].[Revenue]"
        .Update()
    End With
    'Add Revenue column as a measure
    measure = model.Measures.Add()
    With Measure
        .AssociateWith(column, xtrAssociationRole.trAssociationSource)
        .Update()
    End With
    'Auto Design, build cube, and save model
    With model
        .Name = "Sales Revenue"
        .DoAutoDesign()
        .TestBuild(20, True)
        .Update()
        .SaveAs("GO Data Warehouse (query).mdl")
        .Close()
    End With
    objTransApp = Nothing
End Sub

```



---

## Remarques

Le présent document a été développé pour des produits et des services proposés aux Etats-Unis et 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.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services 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 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 utilisant un jeu de caractères double octet peuvent être obtenues par écrit auprès d'IBM à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

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 CONTREFACON AINSI QU'EN CAS DE DEFAULT

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 de l'ICA, des Conditions internationales d'utilisation des 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. 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.

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.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Selon la configuration déployée, la présente Offre Logiciels peut utiliser des cookies de session et des cookies persistants destinés à collecter

- le nom
- le nom utilisateur
- le mot de passe

à des fins

- de gestion de session
- d'authentification
- de facilité d'utilisation des produits
- de configuration d'un code d'accès unique
- de suivi de l'utilisation, ou pour des fonctions autres que celles-ci.

Ces cookies ne peuvent pas être désactivés.

Si les configurations déployées pour cette offre logicielle vous fournissent, en tant que client, la possibilité de collecter des informations identifiant l'utilisateur final via des cookies ou d'autres technologies, vous devez vérifier auprès de votre conseiller juridique les lois applicables en matière de collecte de données, y compris concernant vos obligations d'information préalable et de consentement.

Pour plus d'informations à propos de l'utilisations de nombreuses technologies, y compris de celle des cookies, voir les règles de confidentialité IBM dans <http://www.ibm.com/privacy> ainsi que la politique sur la protection des renseignements personnels IBM en ligne dans <http://www.ibm.com/privacy/details> dans la section appelée "Cookies, balises Web et autres technologies" et les "Règles de confidentialité de produits logiciels et logiciels sous forme de services IBM" dans <http://www.ibm.com/software/info/product-privacy>.

---

## Marques

IBM, le logo IBM et [ibm.com](http://www.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 d'IBM est disponible sur la page Web « Copyright and trademark information » à [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Les termes qui suivent 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.
- Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.
- UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

