

Guide de génération de scripts et
d'automatisation de IBM SPSS Modeler
15



Remarque : Avant d'utiliser ces informations et le produit qu'elles concernent, lisez les informations générales sous Remarques sur p. .

Cette version s'applique à IBM SPSS Modeler 15 et à toutes les publications et modifications ultérieures jusqu'à mention contraire dans les nouvelles versions.

Les captures d'écran des produits Adobe sont reproduites avec l'autorisation de Adobe Systems Incorporated.

Les captures d'écran des produits Microsoft sont reproduites avec l'autorisation de Microsoft Corporation.

Matériel sous licence - Propriété d'IBM

© **Copyright IBM Corporation 1994, 2012.**

Droits limités pour les utilisateurs au sein d'administrations américaines : utilisation, copie ou divulgation soumise au GSA ADP Schedule Contract avec IBM Corp.

Préface

IBM® SPSS® Modeler est le puissant utilitaire de Data mining de IBM Corp.. SPSS Modeler aide les entreprises et les organismes à améliorer leurs relations avec les clients et les citoyens grâce à une compréhension approfondie des données. A l'aide des connaissances plus précises obtenues par le biais de SPSS Modeler, les entreprises et les organismes peuvent conserver les clients rentables, identifier les opportunités de vente croisée, attirer de nouveaux clients, détecter les éventuelles fraudes, réduire les risques et améliorer les services gouvernementaux.

L'interface visuelle de SPSS Modeler met à contribution les compétences professionnelles de l'utilisateur, ce qui permet d'obtenir des modèles prédictifs plus efficaces et de trouver des solutions plus rapidement. SPSS Modeler dispose de nombreuses techniques de modélisation, telles que les algorithmes de prévision, de classification, de segmentation et de détection d'association. Une fois les modèles créés, l'utilisateur peut utiliser IBM® SPSS® Modeler Solution Publisher pour les remettre aux responsables, où qu'ils se trouvent dans l'entreprise, ou pour les transférer vers une base de données.

A propos de IBM Business Analytics

Le logiciel IBM Business Analytics fournit des informations complètes, cohérentes et précises que les preneurs de décision utilisent avec confiance pour améliorer la performance du marché. Un portefeuille étendu d'outils de [business intelligence](#), d'[analyses prédictives](#), de [performance financière et de gestion de stratégie](#), et des [applications analytiques](#) offre des connaissances claires, immédiates et applicables pour améliorer l'efficacité actuelle ainsi que la capacité de prévoir les résultats futurs. Combinées avec de riches solutions industrielles, des pratiques éprouvées et des services professionnels, les organisations de toutes tailles peuvent atteindre la productivité la plus élevée, automatiser des décisions en toute tranquillité et fournir de meilleurs résultats.

Dans le cadre de ce portefeuille, le logiciel IBM SPSS Predictive Analytics aide les organisations à prévoir des événements futurs et à agir en conséquence pour mener à de meilleurs résultats. Des clients dans le domaine commercial, gouvernemental et académique à travers le monde font confiance à la technologie IBM SPSS et considèrent qu'elle représente un avantage compétitif pour attirer, retenir et ajouter des clients, tout en réduisant la fraude et en atténuant les risques. En incorporant le logiciel IBM SPSS dans leur opérations quotidiennes, les organisations deviennent des entreprises prédictives – capables de diriger et d'automatiser les décisions pour atteindre les buts qu'ils se sont fixés et obtenir des avantages compétitifs sensibles. Pour informations supplémentaires ou pour joindre un revendeur, visitez le site <http://www.ibm.com/spss>.

Assistance technique

L'assistance technique est à la disposition des clients pour la maintenance des produits. Les clients peuvent contacter l'assistance technique pour obtenir de l'aide concernant l'utilisation des produits IBM Corp. ou l'installation dans l'un des environnements matériels pris en charge. Pour joindre l'assistance technique, consultez le site Web de IBM Corp. à l'adresse <http://www.ibm.com/support>. Lorsque vous contactez l'assistance technique, n'oubliez pas de préparer vos identifiants, le nom de votre société et votre contrat d'assistance.

Contenu

1 A propos de IBM SPSS Modeler 1

À propos de IBM SPSS Modeler	1
Produits IBM SPSS Modeler	1
IBM SPSS Modeler	1
IBM SPSS Modeler Server	2
IBM SPSS Modeler Administration Console	2
IBM SPSS Modeler Batch	2
IBM SPSS Modeler Solution Publisher	3
IBM SPSS Modeler Server Adaptateurs pour IBM SPSS Collaboration and Deployment Services	3
Éditions de IBM SPSS Modeler	3
Documentation de IBM SPSS Modeler	4
Documentation de SPSS Modeler Professional	4
Documentation de SPSS Modeler Premium	5
Exemples d'application	6
Dossier Demos	6

Partie I: Génération de scripts et langage de script

2 Génération de scripts - Présentation 9

Types de script	9
Scripts de flux	10
Exemple de script de flux : Apprentissage d'un réseau de neurones	11
Scripts autonomes	12
Exemple de script autonome : Enregistrement et chargement d'un modèle	13
Exemple de script autonome : Génération d'un modèle Sélection de fonction	14
Scripts de super noeud	15
Exemple de script de super noeud	16
Exécution et interruption de scripts	16
Rechercher et remplacer	17

3 Langage de script 20

Présentation du langage de script	20
Syntaxe pour les scripts	20

Référence aux noeuds	21
Extraction d'objets	23
Définition de l'objet actuel	23
Ouverture de flux et d'autres objets	24
Utilisation de plusieurs flux.	25
Variables de script locales	25
Paramètres de flux, de session et de super noeud	26
Contrôle de l'exécution du script.	28
Opérateurs de scripts	28
Expressions CLEM dans les scripts.	29
Insertion de commentaires et de lignes incomplètes.	29
Blocs de texte littéral	30

4 Commandes de script

32

Commandes de script générales.	32
execute_all	32
execute_script	32
exit	32
for...endfor	33
if...then...else....	34
Commande set.	34
Commande var	38
Objets de noeud	38
create NODE	38
connect NODE.	39
delete NODE	40
disable NODE	40
disconnect NODE	40
duplicate NODE	40
enable NODE.	40
execute NODE	41
export NODE as FILE	41
flush NODE	42
get node NODE	42
load node FILENAME.	42
position NODE	42
rename NODE as NEWNAME	43
retrieve node REPOSITORY_PATH	43
save node NODE as FILENAME	44
store node NODE as REPOSITORY_PATH.	44

Objets de modèle	44
Noms des nuggets de modèle	44
Pour éviter les noms de modèle en double	46
delete model MODEL	47
export model MODEL as FILE	47
insert model MODEL	49
load model FILENAME	49
retrieve model REPOSITORY_PATH	49
save model MODEL as FILENAME	49
store model MODEL as REPOSITORY_PATH	49
Objets de flux	50
create stream DEFAULT_FILENAME	50
close STREAM	50
clear stream	51
get stream STREAM	51
load stream FILENAME	51
open stream FILENAME	51
retrieve stream REPOSITORY_PATH	51
save STREAM as FILENAME	52
store stream as REPOSITORY_PATH	52
with stream STREAM	53
Objets de projet	53
execute_project	53
load project FILENAME	54
retrieve project REPOSITORY_PATH	54
save project as FILENAME	54
store project as REPOSITORY_PATH	54
Objets d'état	54
load state FILENAME	54
Objets de résultat	55
value RESULT	55
Objets de fichier	55
close FILE	55
open FILE	55
write FILE	56
Objets de sortie	56
Nom des types de sortie	56
delete output OUTPUT	57
export output OUTPUT	57
get output OUTPUT	57
load output FILENAME	58
retrieve output REPOSITORY_PATH	58

save output OUTPUT as FILENAME	58
store output OUTPUT as REPOSITORY_PATH	58

5 *Conseils pour la génération de scripts* **59**

Modification de l'exécution du flux	59
Bouclage dans les noeuds	59
Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository	60
Génération d'un mot de passe codé	62
Vérification du script	62
Génération de scripts à partir de la ligne de commande	63
Compatibilité avec les versions précédentes	63

6 *Exemples de génération de scripts* **65**

Rapport de noeud Typer	65
Rapport de flux	68

7 *Arguments de ligne de commande* **71**

Appel du logiciel	71
Utilisation d'arguments de ligne de commande	71
Combinaison de plusieurs arguments	72
Arguments de connexion au serveur	73
IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion	74
Arguments système	75
Arguments de paramètre	76

8 *Référence du langage CLEM* **77**

Présentation de la référence CLEM	77
Types de données CLEM	77
Entiers	78
Réels	78
Caractères	79
Chaînes	79

Listes	79
Champs	79
Dates	80
Temps	81
Opérateurs CLEM	81
Informations de référence sur les fonctions	83
Conventions pour la description des fonctions	85
Fonctions d'informations	85
Fonctions de conversion	86
Fonctions de comparaison	87
Fonctions logiques	89
Fonctions numériques	90
Fonctions trigonométriques	91
Fonctions de probabilité	92
Opérations d'entiers sur les bits	92
Fonctions aléatoires	94
Fonctions sur chaînes	94
Fonctions SoundEx	99
Fonctions date et heure	100
Fonctions séquentielles	104
Fonctions globales	110
Traitement des valeurs nulles et non renseignées	111
Champs spéciaux	112

Partie II: Référence sur les propriétés

9 Référence sur les propriétés 115

Introduction aux références sur les propriétés	115
Syntaxe des propriétés	115
Exemple de propriétés de noeud et de flux	117
Présentation des propriétés de noeud	118
Propriétés communes des noeuds	118

10 Propriétés du flux **120**

11 Propriétés du projet **123**

12 Propriétés des nœuds source **124**

Propriétés communes aux nœuds source	124
propriétés de cognosimportnode	126
Propriétés de databasenode	128
Propriétés de datacollectionimportnode	130
Propriétés de excelimportnode	132
Propriétés de evimportnode	133
Propriétés de fixedfilenode	134
Propriétés de sasimportnode	136
Propriétés de statisticsimportnode	137
Propriétés de userinputnode	137
Propriétés de variablefilenode	138
Propriétés de xmlimportnode	141

13 Propriétés des noeuds d'opérations sur les lignes **143**

Propriétés de appendnode	143
Propriétés de aggregatenode	143
Propriétés de balancenode	144
Propriétés de distinctnode	145
Propriétés de mergenode	146
Propriétés de rfmaggregatenode	147
Propriétés de samplenode	149
Propriétés de selectnode	151
Propriétés de sortnode	151

14 Propriétés des nœuds d'opérations sur les champs **153**

Propriétés de anonymizenode	153
Propriétés autodatapreprenode	154
Propriétés de binningnode	157
Propriétés de derivenode	159
Propriétés de ensemblenode	161
Propriétés de fillernode	162
Propriétés de filternode	163
Propriétés de historynode	164
Propriétés de partitionnode	165
Propriétés de reclassifynode	166
Propriétés de reordernode	167
Propriétés de restructurenode	168
Propriétés de rfanalysisnode	169
Propriétés de settoflagnode	170
Propriétés statisticstransformnode	171
Propriétés de timeintervalsnode	171
Propriétés de transposenode	176
Propriétés de typenode	176

15 Propriétés des noeuds Graphiques **182**

Propriétés communes aux noeuds Graphiques	182
Propriétés de collectionnode	183
Propriétés de distributionnode	184
Propriétés de evaluationnode	185
Propriétés de graphboardnode	187
Propriétés de histogramnode	189
Propriétés de multiplotnode	190
Propriétés de plotnode	191
Propriétés de timeplotnode	193
Propriétés de webnode	195

16 Propriétés des noeuds de modélisation

197

Propriétés communes des noeuds de modélisation	197
Propriétés de anomalydetectionnode	198
Propriétés de apriorinode	199
Propriétés autoclassifiernode	201
Définition des propriétés de l’algorithme	202
Propriétés du noeud de classification automatique	203
Propriétés de autonumericnode	205
Propriétés de bayesnetnode	206
Propriétés de c50node	208
Propriétés de carmanode	209
Propriétés de cartnode	210
Propriétés de chaidnode	213
Propriétés de coxregnode	215
Propriétés de decisionlistnode	217
Propriétés de discriminantnode	218
Propriétés de factornode	220
Propriétés de featureselectionnode	222
Propriétés de genlinnode	224
Propriétés glmmnode	227
Propriétés de kmeansnode	230
Propriétés knnnode	231
Propriétés de kohonennode	233
Propriétés de linearnode	234
Propriétés de logregnode	235
Propriétés de neuralnetnode	240
Propriétés de neuralnetworknode	242
Propriétés de questnode	244
Propriétés de regressionnode	246
Propriétés de sequencenode	248
Propriétés de slrmnode	249
Propriétés statisticsmodelnode	250
Propriétés de svmnode	250
Propriétés de timeseriesnode	251
Propriétés de twostepnode	254

17 Propriétés du noeud de nugget de modèle 255

Propriétés de applyanomalydetectionnode	255
Propriétés de applypriorinode	255
Propriétés de applyautoclassifiernode	256
Propriétés de applyautoclusternode	256
Propriétés de applyautonumericnode	257
Propriétés de applybayesnetnode	257
Propriétés de applyc50node	257
Propriétés de applycarmanode	258
Propriétés de applycartnode	258
Propriétés de applychaidnode	258
Propriétés de applycoxregnode	259
Propriétés de applydecisionlistnode	259
Propriétés de applydiscriminantnode	259
Propriétés de applyfactornode	260
Propriétés de applyfeatureselectionnode	260
Propriétés de applygeneralizedlinearnode	260
Propriétés de applykmeansnode	260
Propriétés applyknnnode	261
Propriétés de applykohonennode	261
Propriétés de applylinearnode	261
Propriétés de applylogregnode	261
Propriétés de applyneuralnetnode	262
Propriétés de applyneuralnetworknode	262
Propriétés de applyquestnode	263
Propriétés de applyregressionnode	263
Propriétés de applyselflearningnode	263
Propriétés de applysequencenode	264
Propriétés de applysvmnode	264
Propriétés de applytimeseriesnode	264
Propriétés de applytwostepnode	264

18 Propriétés du nœud de modélisation SGBD 265

Propriétés du nœud pour la modélisation Microsoft	266
Propriétés des nœuds de modélisation Microsoft	266
Propriétés du nugget de modèle Microsoft	268

Propriétés du noeud pour la modélisation Oracle	270
Propriétés du noeud de modélisation Oracle	270
Propriétés du nugget de modèle Oracle	275
Propriétés de nœud pour la modélisation IBM DB2	276
Propriétés du noeud de modélisation IBM DB2	276
Propriétés du nugget de modèle IBM DB2	281
Propriétés du noeud pour la modélisation IBM Netezza Analytics	282
Propriétés des nœuds de modélisation Netezza	282
Propriétés du nugget de modèle Netezza	290

19 Propriétés des noeuds de sortie 291

Propriétés de analysisnode	291
Propriétés de dataauditnode	292
Propriétés de matrixnode	294
Propriétés de meansnode	296
Propriétés de reportnode	297
Propriétés de setglobalsnode	298
Propriétés de statisticsnode	299
Propriétés de statisticsoutputnode	300
Propriétés de tablnode	300
Propriétés de transformnode	303

20 Propriétés du nœud d'exportation 305

Propriétés communes des nœuds Exportation	305
Propriétés cognosexportnode	305
Propriétés de databaseexportnode	306
Propriétés de datacollectionexportnode	311
Propriétés de excelexportnode	312
Propriétés de outputfilenode	312
Propriétés de sasexportnode	313
Propriétés de statisticsexportnode	314
Propriétés de xmlexportnode	314

21 Propriétés de noeuds IBM SPSS Statistics 316

Propriétés de statisticsimportnode 316
Propriétés statistictransformnode 316
Propriétés statisticsmodelnode 317
Propriétés de statisticsoutputnode 318
Propriétés de statisticsexportnode 318

22 Propriétés du super noeud 320

Annexe

A Remarques 322

Index 325

A propos de IBM SPSS Modeler

À propos de IBM SPSS Modeler

IBM® SPSS® Modeler est un ensemble d'outils de data mining qui vous permet de développer rapidement, grâce à vos compétences professionnelles, des modèles prédictifs et de les déployer dans des applications professionnelles afin de faciliter la prise de décision. Conçu autour d'un modèle confirmé, le modèle CRISP-DM, SPSS Modeler prend en charge l'intégralité du processus de Data mining, des données à l'obtention de meilleurs résultats commerciaux.

SPSS Modeler propose différentes méthodes de modélisation issues des domaines de l'apprentissage automatique, de l'intelligence artificielle et des statistiques. Les méthodes disponibles dans la palette Modélisation vous permettent d'extraire de nouvelles informations de vos données et de développer des modèles prédictifs. Chaque méthode possède ses propres avantages et est donc plus adaptée à certains types de problème spécifiques.

Il est possible d'acquérir SPSS Modeler comme produit autonome ou de l'utiliser en tant que client en combinaison avec SPSS Modeler Server. Plusieurs autres options sont également disponibles, telles que décrites dans les sections suivantes. Pour plus d'informations, consultez <http://www.ibm.com/software/analytics/spss/products/modeler/>.

Produits IBM SPSS Modeler

La famille des produits IBM® SPSS® Modeler et les logiciels associés sont composés des éléments suivants.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adaptateurs pour IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler est une version complète du produit que vous installez et exécutez sur votre ordinateur personnel. Pour obtenir de meilleures performances lors du traitement d'ensembles de données volumineux, vous pouvez exécuter SPSS Modeler en mode local, comme produit autonome, ou l'utiliser en mode réparti, en association avec IBM® SPSS® Modeler Server.

Avec SPSS Modeler, vous pouvez créer des modèles prédictifs précis rapidement et de manière intuitive, sans aucune programmation. L'interface visuelle unique vous permet de visualiser facilement le processus de Data mining. Grâce aux analyses avancées intégrées au produit, vous pouvez découvrir des motifs et tendances masqués dans vos données. Vous pouvez modéliser les résultats et comprendre les facteurs qui les influencent, afin d'exploiter les opportunités commerciales et de réduire les risques.

SPSS Modeler est disponible en deux éditions : SPSS Modeler Professional et SPSS Modeler Premium. [Pour plus d'informations, reportez-vous à la section Éditions de IBM SPSS Modeler dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

IBM SPSS Modeler Server

Grâce à une architecture client/serveur, SPSS Modeler adresse les demandes d'opérations très consommatrices de ressources à un logiciel serveur puissant. Il offre ainsi des performances accrues sur des ensembles de données plus volumineux.

SPSS Modeler Server est un produit avec licence distincte qui s'exécute en permanence en mode d'analyse réparti sur un hôte de serveur en combinaison avec une ou plusieurs installations de IBM® SPSS® Modeler. Ainsi, SPSS Modeler Server fournit des performances supérieures sur de grands ensembles de données car les opérations nécessitant beaucoup de mémoire peuvent être effectuées sur le serveur sans télécharger de données sur l'ordinateur client. IBM® SPSS® Modeler Server prend également en charge l'optimisation SQL et propose des fonctionnalités de modélisation dans la base de données pour des performances et une automatisation améliorées.

IBM SPSS Modeler Administration Console

Le Modeler Administration Console est une application graphique permettant de gérer de nombreuses options de SPSS Modeler Server qui peuvent également être configurées au moyen d'un fichier d'options. Cette application offre une interface utilisateur sous forme de console permettant de surveiller et de configurer les installations SPSS Modeler Server ; elle est disponible gratuitement pour les clients actuels de SPSS Modeler Server. L'application ne peut être installée que sur des ordinateurs Windows ; en revanche, elle peut administrer un serveur installé sur n'importe quelle plate-forme prise en charge.

IBM SPSS Modeler Batch

Alors que le Data mining est généralement un processus interactif, il est également possible d'exécuter SPSS Modeler à partir d'une ligne de commande sans recourir à l'interface utilisateur graphique. Par exemple, vous pouvez avoir des tâches longue durée ou répétitives à exécuter sans intervention de l'utilisateur. SPSS Modeler Batch est une version spécifique du produit qui prend en charge toutes les fonctions d'analyse de SPSS Modeler sans avoir besoin d'accéder à l'interface utilisateur standard. Une licence SPSS Modeler Server est nécessaire pour utiliser SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher est un outil qui permet de créer une version « packagée » d'un flux SPSS Modeler qui peut être exécutée par un moteur Runtime externe ou intégrée dans une application externe. Ainsi, vous pouvez publier et déployer des flux SPSS Modeler complets dans des environnements où SPSS Modeler n'est pas installé. SPSS Modeler Solution Publisher est fourni avec le service IBM SPSS Collaboration and Deployment Services - Scoring et nécessite une licence distincte. Avec cette licence, vous recevez SPSS Modeler Solution Publisher Runtime qui vous permet d'exécuter les flux publiés.

IBM SPSS Modeler Server Adaptateurs pour IBM SPSS Collaboration and Deployment Services

Différents adaptateurs pour IBM® SPSS® Collaboration and Deployment Services sont disponibles et permettent à SPSS Modeler et SPSS Modeler Server d'interagir avec un référentiel IBM SPSS Collaboration and Deployment Services. Ainsi, un flux SPSS Modeler déployé sur le référentiel peut être partagé par différents utilisateurs ou peut être accessible depuis l'application client léger IBM SPSS Modeler Advantage. Installez l'adaptateur sur le système qui héberge le référentiel.

Éditions de IBM SPSS Modeler

SPSS Modeler est disponible dans les éditions suivantes.

SPSS Modeler Professional

SPSS Modeler Professional offre tous les outils nécessaires à l'utilisation de la plupart des types de données structurées, tels que les comportements et interactions suivis dans les systèmes CRM, les caractéristiques sociodémographiques, les comportements d'achat et les données de vente.

SPSS Modeler Premium

SPSS Modeler Premium est un produit avec licence distincte qui étend le champ d'applications de SPSS Modeler Professional afin de pouvoir traiter des données spécialisées telles que celles utilisées pour les analyses d'entités ou les réseaux sociaux ainsi que des données de texte non structurées. SPSS Modeler Premium comprend les composants suivants :

IBM® SPSS® Modeler Entity Analytics ajoute une dimension entièrement nouvelle aux analyses prédictives IBM® SPSS® Modeler. Alors que les analyses prédictives essaient de prévoir les comportements futurs à partir de données passées, les analyses d'entités se concentrent sur l'amélioration de la cohérence des données actuelles en résolvant les conflits d'identités dans les enregistrements eux-mêmes. Une identité peut être celle d'un individu, d'une organisation, d'un objet ou d'une autre entité pour laquelle une ambiguïté peut exister. La résolution d'identité peut être vitale dans de nombreux domaines, y compris la gestion de la relation client, la détection de la fraude, le blanchiment d'argent et la sécurité nationale et internationale.

IBM SPSS Modeler Social Network Analysis transforme les informations sur les relations en champs qui caractérisent le comportement social des individus et des groupes. Grâce aux données qui décrivent les relations qui sous-tendent les réseaux sociaux, IBM® SPSS® Modeler Social Network Analysis identifie les chefs sociaux qui influencent le comportement des autres individus du réseau. De plus, il est possible de déterminer les individus qui sont le plus influencés par les autres participants du réseau. En combinant ces résultats avec d'autres mesures, il est possible de créer des profils détaillés des individus sur lesquels baser vos modèles prédictifs. Les modèles qui contiennent ces informations sociales seront plus efficaces que les modèles qui en sont dépourvus.

Text Analytics for IBM® SPSS® Modeler utilise des technologies linguistiques avancées et le traitement du langage naturel pour traiter rapidement une large variété de données textuelles non structurées, en extraire les concepts clés et les organiser pour les regrouper dans des catégories. Les concepts extraits et les catégories peuvent ensuite être combinés aux données structurées existantes, telles que les données démographiques, et appliqués à la modélisation grâce à la gamme complète d'outils de Data mining de SPSS Modeler, afin de favoriser une prise de décision précise et efficace.

Documentation de IBM SPSS Modeler

Une documentation au format d'aide en ligne est disponible dans le menu Aide de SPSS Modeler. Vous y trouverez la documentation de SPSS Modeler, SPSS Modeler Server et de SPSS Modeler Solution Publisher, ainsi que le Guide des applications et d'autres documentations utiles.

La documentation complète de chaque produit (y compris les instructions d'installation) au format PDF est disponible dans le dossier *Documentation* de chaque DVD de produit. Ces documents d'installation peuvent également être téléchargés sur Internet à l'adresse <http://www-01.ibm.com/support/docview.wss?uid=swg27023172>.

La documentation dans les deux formats est également disponible depuis le Centre d'informations SPSS Modeler à l'adresse <http://publib.boulder.ibm.com/infocenter/spssmodl/v15r0m0/>.

Documentation de SPSS Modeler Professional

La suite de documentation SPSS Modeler Professional (à l'exception des instructions d'installation) est la suivante.

- **Guide de l'utilisateur IBM SPSS Modeler.** Introduction générale à SPSS Modeler : création de flux de données, traitement des valeurs manquantes, création d'expressions CLEM, utilisation des projets et des rapports et regroupement des flux pour le déploiement dans IBM SPSS Collaboration and Deployment Services, des applications prédictives ou IBM SPSS Modeler Advantage.
- **Noeuds de Source, d'exécution et de sortie IBM SPSS Modeler.** Descriptions de tous les noeuds utilisés pour lire, traiter et renvoyer les données de sortie dans différents formats. En pratique, cela signifie tous les noeuds autres que les noeuds de modélisation.
- **IBM SPSS Modeler Noeuds de modélisation.** Description de tous les noeuds utilisés pour créer des modèles de Data mining. IBM® SPSS® Modeler propose différentes méthodes de modélisation issues des domaines de l'apprentissage automatique, de l'intelligence artificielle

et des statistiques. [Pour plus d'informations, reportez-vous à la section Description des noeuds de modélisation dans le chapitre 3 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

- **Guide des Algorithmes IBM SPSS Modeler.** Descriptions des fondements mathématiques des méthodes de modélisation utilisées dans SPSS Modeler. Ce guide est disponible au format PDF uniquement.
- **Guide des applications IBM SPSS Modeler.** Les exemples de ce guide fournissent des introductions brèves et ciblées aux méthodes et techniques de modélisation. Une version en ligne de ce guide est également disponible dans le menu Aide. [Pour plus d'informations, reportez-vous à la section Exemples d'application dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)
- **Génération de scripts et automatisation IBM SPSS Modeler.** Informations sur l'automatisation du système via la génération de scripts, y compris les propriétés permettant de manipuler les noeuds et les flux.
- **IBM SPSS Modeler Guide de déploiement.** Informations sur l'exécution des scénarios et des flux SPSS Modeler comme étapes des tâches d'exécution sous IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **IBM SPSS Modeler CLEF Guide du développeur.** CLEF permet d'intégrer des programmes tiers tels que des programmes de traitement de données ou des algorithmes de modélisation en tant que noeuds dans SPSS Modeler.
- **Guide d'exploration de base de données IBM SPSS Modeler.** Informations sur la manière de tirer parti de la puissance de votre base de données pour améliorer les performances et étendre la gamme des fonctions analytiques via des algorithmes tiers.
- **Guide des performances et d'administration IBM SPSS Modeler Server.** Informations sur le mode de configuration et d'administration de IBM® SPSS® Modeler Server.
- **Guide de l'utilisateur de IBM SPSS Modeler Administration Console.** Informations concernant l'installation et l'utilisation de l'interface utilisateur de la console permettant de surveiller et de configurer SPSS Modeler Server. La console est implémentée en tant que plug-in à l'application Deployment Manager.
- **Guide IBM SPSS Modeler Solution Publisher.** SPSS Modeler Solution Publisher est un module complémentaire qui permet aux entreprises de publier des flux destinés à être utilisés en dehors de l'environnement SPSS Modeler.
- **Guide CRISP-DM IBM SPSS Modeler** Guide détaillé sur l'utilisation de la méthodologie CRISP-DM pour le Data mining avec SPSS Modeler
- **Guide de l'utilisateur IBM SPSS Modeler Batch.** Guide complet sur l'utilisation de IBM SPSS Modeler en mode par lots, avec des détails sur l'exécution en mode par lots et les arguments de ligne de commande. Ce guide est disponible au format PDF uniquement.

Documentation de SPSS Modeler Premium

La suite de documentation SPSS Modeler Premium (à l'exception des instructions d'installation) est la suivante.

- **IBM SPSS Modeler Entity Analytics Guide de l'utilisateur.** Informations sur l'utilisation des analyses d'entités avec SPSS Modeler, notamment l'installation et la configuration du référentiel, les noeuds d'analyses d'entités et les tâches administratives.

- **IBM SPSS Modeler Social Network Analysis Guide de l'utilisateur.** Guide sur l'exécution des analyses de réseaux sociaux avec SPSS Modeler, y compris les analyses de groupe et analyses de diffusion.
- **Text Analytics for SPSS Modeler Guide de l'utilisateur.** Informations sur l'utilisation des analyses de texte avec SPSS Modeler, notamment sur les nœuds de Text Mining, l'espace de travail interactif, les modèles et d'autres ressources.
- Guide de l'utilisateur de **Text Analytics for IBM SPSS Modeler Administration Console.** Informations concernant l'installation et l'utilisation de l'interface utilisateur de la console permettant de surveiller et de configurer IBM® SPSS® Modeler Server pour l'utiliser avec Text Analytics for SPSS Modeler. La console est implémentée en tant que plug-in à l'application Deployment Manager.

Exemples d'application

Tandis que les outils de Data mining de SPSS Modeler peuvent vous aider à résoudre une grande variété de problèmes commerciaux et organisationnels, les exemples d'application fournissent des introductions brèves et ciblées aux méthodes et aux techniques de modélisation. Les ensembles de données utilisés ici sont beaucoup plus petits que les énormes entrepôts de données gérés par certains Data miners, mais les concepts et les méthodes impliqués doivent pouvoir être adaptés à des applications réelles.

Vous pouvez accéder aux exemples en cliquant Exemples d'application dans le menu Aide de SPSS Modeler. Les fichiers de données et les flux d'échantillons sont installés dans le dossier *Demos*, sous le répertoire d'installation du produit. [Pour plus d'informations, reportez-vous à la section Dossier Demos dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Exemples de modélisation de bases de données. Consultez les exemples dans le *IBM SPSS ModelerGuide d'exploration de base de données*.

Exemples de génération de scripts. Consultez les exemples dans le *IBM SPSS ModelerGuide de génération de scripts et d'automatisation*.

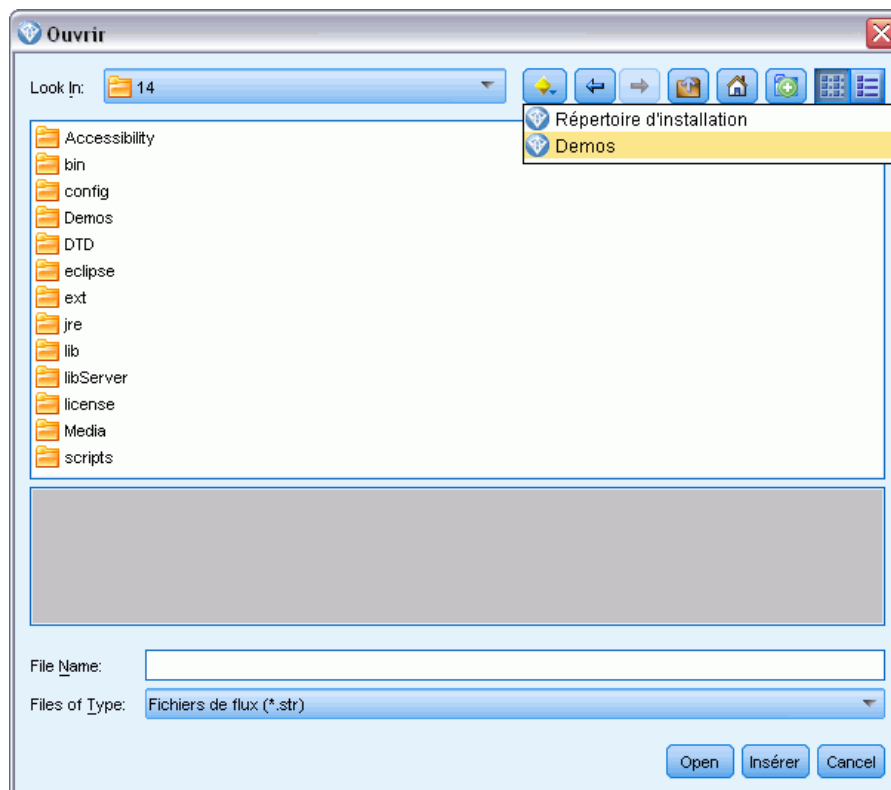
Dossier Demos

Les fichiers de données et les flux d'échantillons utilisés avec les exemples d'application sont installés dans le dossier *Demos*, sous le répertoire d'installation du produit. Ce dossier est également accessible à partir du groupe de programmes sous IBM SPSS Modeler 15 dans le menu

Démarrer de Windows, ou en cliquant sur *Demos* dans la liste des répertoires récents de la boîte de dialogue Ouverture de fichier.

Figure 1-1

Sélection du dossier *Demos* dans la liste des répertoires récemment consultés



Partie I:
Génération de scripts et langage de script

Génération de scripts - Présentation

La génération de scripts IBM® SPSS® Modeler est un outil performant pour automatiser les processus dans l'interface utilisateur. Les scripts permettent d'effectuer les mêmes opérations qu'avec la souris ou le clavier. Vous pouvez les utiliser pour automatiser les tâches dont l'exécution manuelle s'avère très répétitive et très longue.

Vous pouvez utiliser les scripts pour :

- imposer un ordre spécifique d'exécution des noeuds dans un flux.
- Définir des propriétés pour un noeud et effectuer des calculs en utilisant un sous-ensemble du CLEM (Control Language for Expression Manipulation).
- Définir l'exécution automatique d'une séquence d'actions qui nécessite normalement l'intervention de l'utilisateur (création et test d'un modèle, par exemple).
- Mettre en place des processus complexes dans lesquels l'intervention de l'utilisateur est importante (procédures de validation croisée qui exigent la création et le test de nombreux modèles, par exemple).
- Mettre en place des processus de gestion des flux (exécution d'un flux d'apprentissage de modèle et production automatique du flux de test modèle correspondant, par exemple).

Ce dessection offre des descriptions et des exemples de haut niveau de scripts de flux, de scripts autonomes et de scripts dans les super noeuds dans l'interface SPSS Modeler. Des informations sur la syntaxe, les commandes et le langage de script vous sont fournies dans les dessections suivantes.

Remarque : Vous ne pouvez pas importer et exécuter des scripts créés dans IBM® SPSS® Statistics avec SPSS Modeler.

Types de script

IBM® SPSS® Modeler utilise trois types de script :

- **Les scripts de flux** sont stockés en tant que propriété de flux et sont, par conséquent, sauvegardés et chargés avec un flux spécifique. Par exemple, vous pouvez écrire un script de flux qui automatise le processus de formation et d'application d'un nugget de modèle. Vous pouvez également préciser que lorsqu'un script particulier s'exécute, ce script doit être exécuté à la place du contenu de l'espace de travail de flux.
- **Les scripts autonomes** ne sont pas associés à un flux en particulier et sont sauvegardés dans des fichiers texte externes. Par exemple, vous pouvez utiliser un script autonome pour manipuler plusieurs flux ensemble.
- **Les scripts de super noeud** sont stockés en tant que propriété de flux de super noeud. Les scripts de super noeud sont uniquement accessibles dans les super noeuds terminaux. Vous pouvez utiliser un script de super noeud pour contrôler la séquence d'exécution des contenus de super noeuds. Pour les super noeuds qui ne sont pas terminaux (source ou processus),

vous pouvez définir les propriétés du super noeud ou des noeuds qu'il contient directement dans votre script de flux.

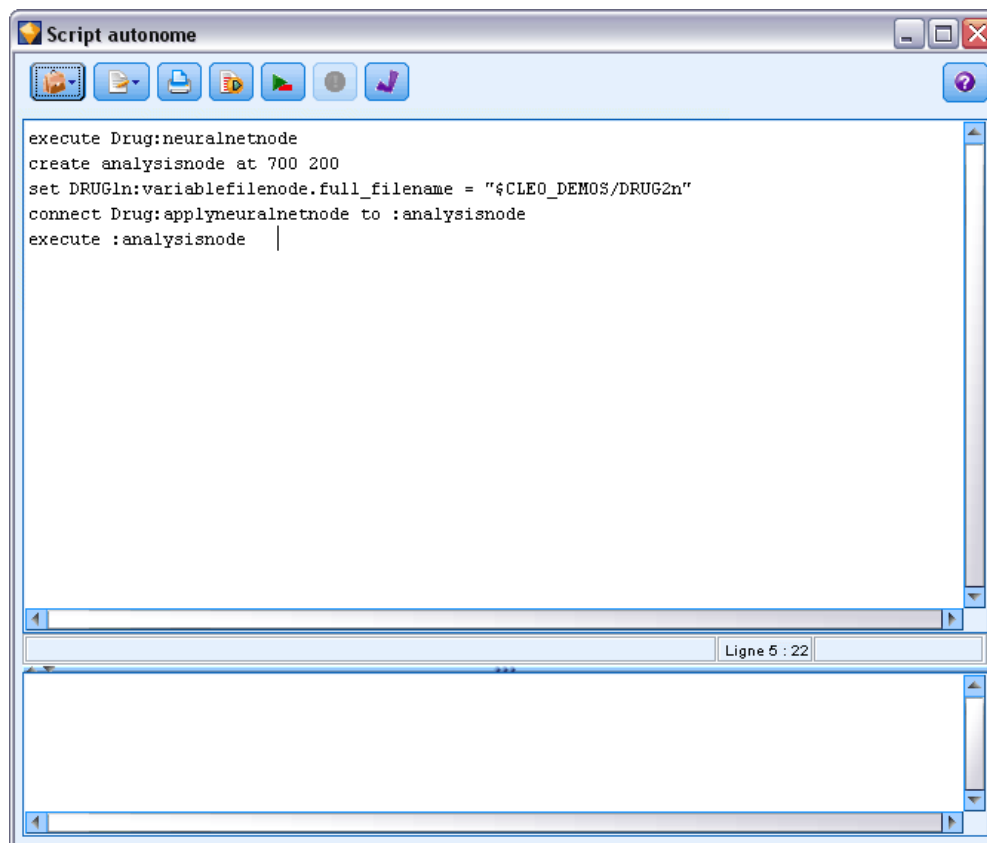
Scripts de flux

Les scripts peuvent être utilisés pour personnaliser les opérations réalisées au sein d'un flux particulier, et ils peuvent être enregistrés avec ce flux. Les scripts de flux peuvent être utilisés pour spécifier un ordre d'exécution particulier des noeuds terminaux dans un flux. La boîte de dialogue du script de flux permet d'éditer le script qui est enregistré avec le flux en cours.

Pour accéder à l'onglet du script de flux dans la boîte de dialogue Propriétés du flux :

- ▶ Dans le menu Outils, sélectionnez :
Propriétés du flux > Script
- ▶ Cliquez sur l'onglet Script pour utiliser les scripts du flux en cours.

Figure 2-1
Boîte de dialogue Script de flux



Les icônes de la barre d'outils se trouvant en haut de cette boîte de dialogue vous permettent d'effectuer les opérations suivantes :

- Importer les contenus d'un script autonome préexistant dans la fenêtre.

- Sauvegarder un script comme fichier texte.
- Imprimer un script.
- Ajouter le script par défaut.
- Exécuter entièrement le script en cours.
- Exécuter les lignes sélectionnées d'un script.
- Vérifier la syntaxe d'un script et, en cas d'erreurs, les afficher dans le panneau inférieur de la boîte de dialogue.

De plus, vous pouvez spécifier si ce script doit ou non être exécuté lorsque le flux est exécuté. Vous pouvez sélectionner Exécuter ce script pour exécuter le script chaque fois que le flux est exécuté en respectant l'ordre d'exécution du script. Ce paramètre permet l'automatisation au niveau du flux et l'accélération de la vitesse de création des modèles. Cependant, le paramètre par défaut est d'ignorer ce script pendant l'exécution du flux. Même si vous sélectionnez l'option Ignorer ce script, vous pouvez quand même exécuter ce script directement depuis la boîte de dialogue.

Exemple de script de flux : Apprentissage d'un réseau de neurones

Lorsqu'il est exécuté, un flux peut être utilisé pour entraîner un modèle de réseau de neurones. La procédure de test normale consiste à exécuter le noeud de modélisation pour ajouter le modèle au flux, définir les connexions appropriées et exécuter un noeud Analyse.

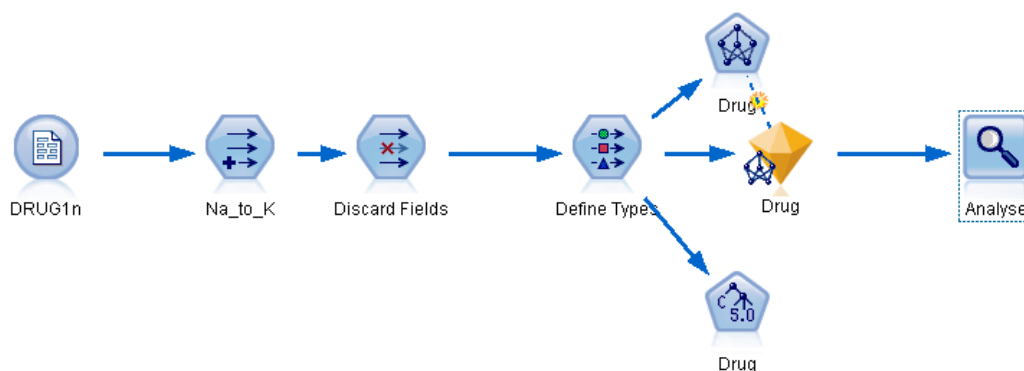
L'utilisation d'un script IBM® SPSS® Modeler permet d'automatiser le processus de test du nugget de modèle après sa création. Par exemple, le script de flux suivant pour tester le flux de démo *druglearn.str* (disponible dans le dossier */Demos/streams/* dans votre installation SPSS Modeler) peut être exécuté dans la boîte de dialogue Propriétés du flux (Outils > Propriétés du flux > Script) :

```
execute Drug:neuralnetworknode
create analysisnode at 700 200
set DRUG1n:variablefilenode.full_filename = "$CLEO_DEMOS/DRUG2n"
connect :applyneuralnetworknode to :analysisnode
execute :analysisnode
```

Les puces suivantes décrivent chaque ligne dans cet exemple de script.

- La première ligne exécute le noeud R. neurones appelé Drug qui se trouve déjà dans le flux de démo afin de générer un nugget de modèle et le place dans l'espace de travail du flux, connecté au noeud Typer déjà présent dans le flux.
- Ligne 2, le script crée un noeud Analyse et le place dans la position 700 X 200 de l'espace de travail.
- Ligne 3, la source de données d'origine utilisée dans le flux est transférée dans un ensemble de données de test appelé DRUG2n.
- Ligne 4, le nugget de modèle Réseau de neurones est connecté au noeud Analyse. Veuillez noter qu'aucun nom n'est utilisé pour signaler le nugget de modèle Réseau de neurones ou le noeud Analyse car aucun autre noeud semblable n'existe dans le flux.
- Finalement, le noeud Analyse est exécuté pour produire le rapport Analyse.

Figure 2-2
Flux généré



Ce script est conçu pour fonctionner avec un flux existant ; en effet, il suppose qu'un noeud R. neurones nommé *Drug* (médicament) existe déjà. Il est également possible d'utiliser un script pour créer intégralement un flux à partir d'un espace de travail vierge, et exécuter ce flux. Pour en savoir plus sur le langage de script en général, consultez Présentation du langage de script sur p. 20. Pour en savoir plus sur des commandes de script spécifiques, consultez Commandes de script sur p. 32.

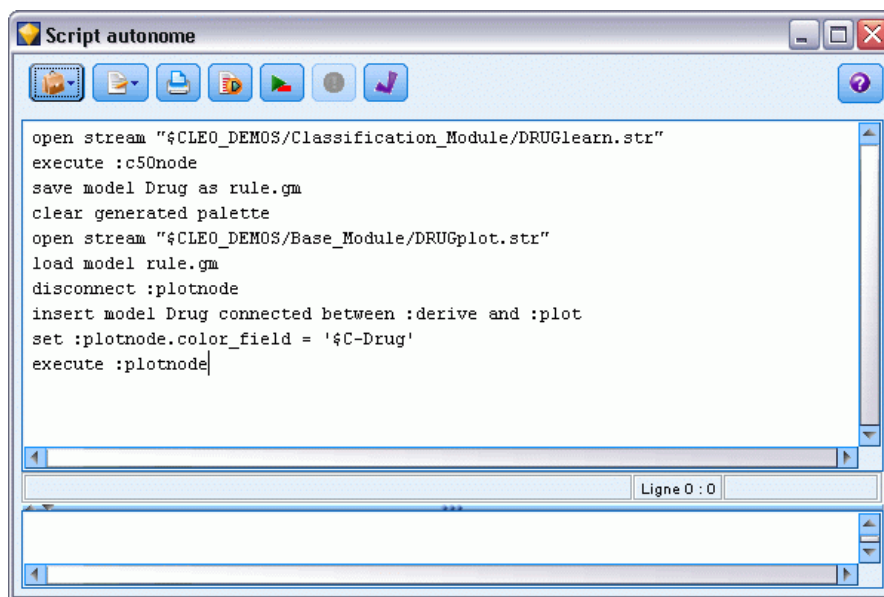
Scripts autonomes

Dans la boîte de dialogue Script autonome, vous pouvez créer ou éditer un script sauvegardé en tant que fichier texte. Cette boîte de dialogue indique le nom du fichier et permet de charger, d'enregistrer, d'importer et d'exécuter des scripts.

Pour accéder à la boîte de dialogue du script autonome :

- Dans le menu principal, sélectionnez :
Outils > Script autonome

Figure 2-3
Boîte de dialogue Script autonome



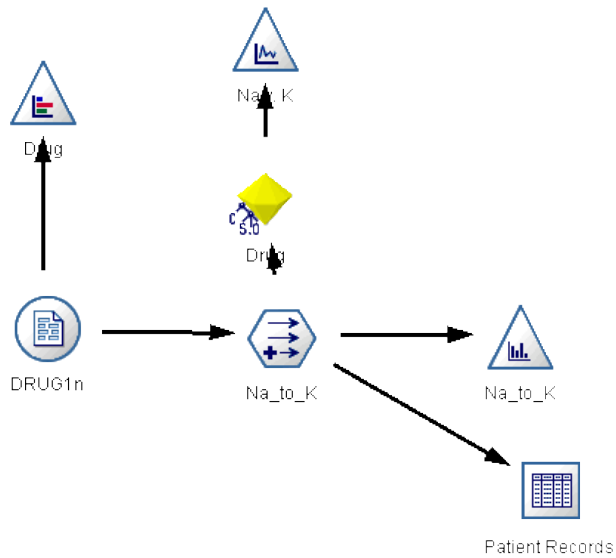
Les scripts autonomes disposent de la même barre d'outils et des mêmes options de vérification de syntaxe de script que les scripts de flux. [Pour plus d'informations, reportez-vous à la section Scripts de flux sur p. 10.](#)

Exemple de script autonome : Enregistrement et chargement d'un modèle

Les scripts autonomes permettent de gérer des flux. Imaginez que vous avez deux flux, l'un qui crée un modèle et l'autre qui utilise les graphiques pour explorer l'ensemble de règles généré à partir du premier flux contenant des champs de données existants. Le script correspondant à ce cas de figure pourrait être semblable au suivant :

```
open stream "$CLEO_DEMOS/streams/druglearn.str"
execute :c50node
save model Drug as rule.gm
clear generated palette
open stream "$CLEO_DEMOS/streams/drugplot.str"
load model rule.gm
disconnect :plotnode
insert model Drug connected between :derive and :plot
set :plotnode.color_field = '$C-Drug'
execute :plotnode
```

Figure 2-4
Flux généré



Remarque : Pour en savoir plus sur le langage de script en général, consultez Présentation du langage de script sur p. 20. Pour en savoir plus sur des commandes de script spécifiques, consultez Commandes de script sur p. 32.

Exemple de script autonome : Génération d'un modèle Sélection de fonction

A partir d'un espace de travail vierge, cet exemple crée un flux qui génère un modèle Sélection de fonction, applique le modèle et crée un tableau répertoriant les 15 champs les plus importants pour la cible indiquée.

```
create stream 'featureselection'
create statisticsimportnode
position :statisticsimportnode at 50 50
set :statisticsimportnode.full_filename = "$CLEO_DEMOS/customer_dbase.sav"

create typenode
position :typenode at 150 50
set :typenode.direction.'response_01' = Target
connect :statisticsimportnode to :typenode

create featureselectionnode
position :featureselectionnode at 250 50
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.important_label = "Check Me Out!"
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
connect :typenode to :featureselectionnode
execute :featureselectionnode
```

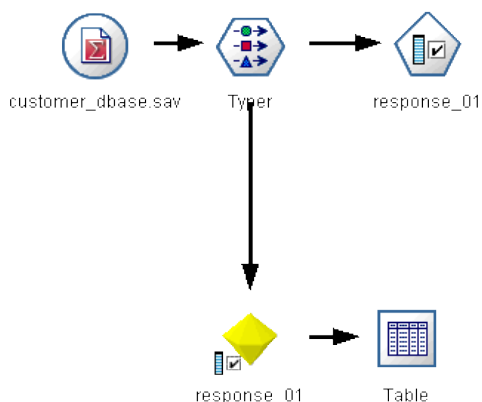
```

create tablenode
position :tablenode at 250 250
connect response_01:applyfeatureselectionnode to :tablenode
execute :tablenode

```

Le script crée un noeud source permettant la lecture des données, utilise un noeud Typer pour affecter le rôle (direction) Target au champ *response_01*, puis crée et exécute un noeud Sélection de fonction. Le script connecte également les noeuds et les positionne sur l'espace de travail de flux afin d'offrir une présentation lisible. Le nugget de modèle résultant est ensuite connecté à un noeud Table qui répertorie les 15 champs les plus importants, en fonction des propriétés *selection_mode* et *top_n*. [Pour plus d'informations, reportez-vous à la section Propriétés de featureselectionnode dans le chapitre 16 sur p. 222.](#)

Figure 2-5
Flux généré



Scripts de super noeud

Vous pouvez créer et sauvegarder des scripts dans tous les super noeuds terminaux avec le langage de script de IBM® SPSS® Modeler. Ces scripts sont uniquement accessibles pour les super noeuds terminaux et sont souvent utilisés lors de la création de modèles de flux ou pour imposer un ordre d'exécution particulier aux contenus des super noeuds. Les scripts de super noeud permettent également d'exécuter plusieurs scripts dans un même flux.

Par exemple, imaginons que vous avez besoin de préciser l'ordre d'exécution d'un flux complexe et que votre super noeud contient plusieurs noeuds qui comprennent un noeud *v. globales* qui doit être exécuté avant de calculer un nouveau champ utilisé dans un noeud Nuage. Dans ce cas, vous pouvez créer un script de super noeud qui exécute d'abord le noeud *v. globales*. Les valeurs calculées par ce noeud, telles que la moyenne ou l'écart-type, peuvent être utilisées lorsque le noeud Nuage est exécuté.

Dans le script de super noeud, vous pouvez spécifier les propriétés du noeud de la même façon que les autres scripts. Ou alors, vous pouvez modifier et définir les propriétés de n'importe quel super noeud ou de ses noeuds encapsulés directement depuis un script de flux. [Pour plus d'informations, reportez-vous à la section Propriétés du super noeud dans le chapitre 22 sur p.](#)

320. Cette méthode fonctionne pour les super noeuds de source et de processus ainsi que pour les super noeuds terminaux.

Remarque : Parce que seuls les super noeuds terminaux peuvent exécuter leurs propres scripts, l'onglet Scripts de la boîte de dialogue Super noeud est uniquement accessible pour les super noeuds terminaux.

Pour ouvrir la boîte de dialogue de script des super noeuds depuis l'espace de travail principal :

- ▶ Sélectionnez un super noeud terminal dans l'espace de travail de flux et, dans le menu Super noeud, choisissez :
Script Super noeud...

Pour ouvrir la boîte de dialogue de script des super noeuds depuis l'espace de travail de super noeud en zoom avant :

- ▶ Cliquez avec le bouton droit de la souris sur l'espace de travail de super noeud et, dans le menu contextuel, sélectionnez :
Script Super noeud...

Pour plus d'informations, reportez-vous à la section Super noeuds et génération de scripts dans le chapitre 9 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.

Exemple de script de super noeud

Le script de super noeud suivant définit l'ordre dans lequel les noeuds terminaux doivent être exécutés à l'intérieur du super noeud. Cet ordre permet que le noeud v. globales soit exécuté en premier afin que les valeurs calculées par ce noeud puissent ensuite être utilisées lorsqu'un autre noeud est exécuté.

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

Exécution et interruption de scripts

Plusieurs manières d'exécuter les scripts sont disponibles. Par exemple, dans la boîte de dialogue du script de flux ou du script autonome, le bouton « Exécuter ce script » permet d'exécuter l'intégralité du script :

Figure 2-6
Bouton Exécuter ce script



Le bouton « Exécuter les lignes sélectionnées » exécute une ligne unique ou un bloc de lignes adjacentes que vous avez sélectionnées dans le script :

Figure 2-7

Bouton Exécuter les lignes sélectionnées



Pour exécuter un script, utilisez l'une des méthodes suivantes :

- Cliquer sur le bouton « Exécuter ce script » ou sur le bouton « Exécuter les lignes sélectionnées » dans la boîte de dialogue du script de flux ou du script autonome.
- Exécutez un flux dans lequel la méthode d'exécution par défaut est Exécuter ce script.
- Utilisez le commutateur `-execute` au démarrage en mode interactif. [Pour plus d'informations, reportez-vous à la section Utilisation d'arguments de ligne de commande dans le chapitre 7 sur p. 71.](#)

Remarque : un script associé à un super noeud est exécuté en même temps que ce dernier si vous avez sélectionné Exécuter ce script dans la boîte de dialogue de script du super noeud.

Interruption de l'exécution d'un script

Dans la boîte de dialogue du script de flux, le bouton rouge d'arrêt est activé pendant l'exécution du script. Ce bouton permet d'abandonner l'exécution du script et de tout flux actuel.

Rechercher et remplacer

La boîte de dialogue Rechercher/Remplacer est disponible aux emplacements où vous modifier un script ou le texte d'une expression, par exemple l'éditeur de script ou le générateur d'expression CLEM, ou encore lors de la définition d'un modèle dans le noeud Rapport. Lorsque vous éditez du texte dans l'un de ces champs, appuyez sur Ctrl-F pour accéder à la boîte de dialogue, en vous assurant que le curseur est centré sur une zone de texte. Si vous travaillez dans un noeud Remplacer, par exemple, vous pouvez accéder à la boîte de dialogue depuis toute zone de texte de l'onglet Paramètres, ou depuis le champ de texte du générateur d'expression.

Figure 2-8

Boîte de dialogue Rechercher/Remplacer



- ▶ Lorsque le curseur se trouve sur une zone de texte, appuyez sur Ctrl+F pour accéder à la boîte de dialogue Rechercher/Remplacer.
- ▶ Entrez le texte que vous souhaitez rechercher ou faites un choix dans la liste déroulante des éléments récemment consultés.
- ▶ Saisissez éventuellement un texte de remplacement.
- ▶ Cliquez sur Suivant pour lancer la recherche.
- ▶ Cliquez sur Remplacer pour remplacer la sélection courante, ou Remplacer tout pour mettre à jour certaines instances ou les instances sélectionnées.
- ▶ La boîte de dialogue se ferme après chaque opération. Appuyez sur F3 depuis toute zone de texte pour répéter la dernière opération de recherche ou sur Ctrl+F pour accéder de nouveau à la boîte de dialogue.

Options de recherche

Respecter la casse. Spécifie si l'opération de recherche est sensible à la casse ; par exemple, si *myvar* correspond à *myVar*. Le texte de remplacement est toujours inséré exactement tel qu'il a été saisi, quel que soit le réglage de ce paramètre.

Mot entier. Spécifie si l'opération de recherche doit porter sur le texte inséré dans des mots. Si cette option est sélectionnée, par exemple, une recherche portant sur *spider* ne produira pas la réponse *spiderman* ou *spider-man*.

Caractères génériques. Spécifie si la syntaxe des caractères génériques est utilisée (voir la section suivante). Lorsqu'elle est sélectionnée, l'option Mot entier est désactivée et sa valeur est ignorée.

Texte sélectionné. Contrôle la portée de la recherche lorsque vous utilisez l'option Remplacer tout.

Syntaxe des caractères génériques

Les caractères génériques vous permettent de rechercher des caractères spéciaux tels que les tabulations ou les sauts de ligne, des classes ou des intervalles de caractères telles que *a* à *d*, toute valeur numérique ou non et les limites telles que le début ou la fin d'une ligne. Les types d'expression suivants sont pris en charge :

Correspondances de caractères

Caractères	Correspondances
x	Le caractère x
\\	Le caractère barre oblique inversée
\\0n	Le caractère présentant la valeur octale 0n (0 <= n <= 7)
\\0nn	Le caractère présentant la valeur octale 0nn (0 <= n <= 7)
\\0mnn	Le caractère présentant la valeur octale 0mnn (0 <= m <= 3, 0 <= n <= 7)
\\xhh	Le caractère présentant la valeur hexadécimale 0xhh
\\uhhhh	Le caractère présentant la valeur hexadécimale 0xhhhh
\\t	Le caractère tabulation ('\\u0009')

Caractères	Correspondances
\n	Le caractère saut de ligne (retour à la ligne) ('u000A')
\r	Le caractère retour chariot ('u000D')
\f	Le caractère alimentation de formulaire ('u000C')
\a	Le caractère alerte (sonnerie) ('u0007')
\e	Le caractère d'échappement ('u001B')
\cx	Le caractère de contrôle correspondant à x

Classes de caractères correspondantes

Classes de caractères	Correspondances
[abc]	a, b ou c (classe simple)
[^abc]	Tout caractère excepté a, b ou c (soustraction)
[a-zA-Z]	Caractères de a à z ou de A à Z compris (intervalle)
[a-d[m-p]]	Caractères de a à d ou de m à p (union). Cette option peut aussi être spécifiée comme [a-dm-p]
[a-z&&[def]]	Caractères de a à z ainsi que d, e ou f (intersection)
[a-z&&[^bc]]	Caractères de a à z à l'exception de b et c (soustraction). Cette option peut aussi être spécifiée comme [ad-z]
[a-z&&[^m-p]]	Caractères de a à z, mais pas de m à p (soustraction). Cette option peut aussi être spécifiée comme [a-lq-z]

Classes de caractères prédéfinies

Classes de caractères prédéfinies	Correspondances
.	Tout caractère (peut correspondre ou non aux terminaisons de ligne)
\d	Tout chiffre : [0-9]
\D	Un caractère non numérique : [^0-9]
\s	Un espace blanc : [\t\n\x0B\f\r]
\S	Un espace non blanc : [^\s]
\w	Un mot : [a-zA-Z_0-9]
\W	Un caractère autre qu'un mot : [^\w]

Correspondances de limites

Correspondances de limites	Correspondances
^	Le début d'une ligne
\$	La fin d'une ligne
\b	Une limite de mot
\B	Une limite autre que celle d'un mot
\A	Le début de la saisie
\Z	La fin de la saisie à l'exception de la terminaison finale éventuelle
\z	La fin de la saisie

Langage de script

Présentation du langage de script

Le langage de script IBM® SPSS® Modeler est constitué des éléments suivants :

- un format pour le référencement des noeuds, des flux, des projets, des sorties, ainsi que d'autres objets SPSS Modeler ;
- Un ensemble d'instructions ou de commandes de script qui permettent de manipuler ces objets.
- Un langage d'expression de script pour le paramétrage des valeurs des variables, paramètres et autres objets.
- Une prise en charge des commentaires, des lignes incomplètes et des blocs de texte littéral.

Cette section décrit la syntaxe de base qui s'applique au langage de script. Des informations sur les propriétés et commandes spécifiques vous sont fournies dans les sections suivantes.

Syntaxe pour les scripts

Pour plus de clarté lors de l'analyse, respectez les règles suivantes quand vous utilisez des scripts dans IBM® SPSS® Modeler :

- Les noms de variable comme `income` ou `referrerID` ne doivent pas être placés entre guillemets.
- Les noms de variable, comme `^mystream` (`^monflux`), sont précédés du caractère « ^ » lorsqu'ils font référence à une variable existante dont la valeur a déjà été définie. Toutefois, ce caractère n'est pas utilisé lors de la déclaration ou de la définition de la valeur de la variable. [Pour plus d'informations, reportez-vous à la section Référence aux noeuds sur p. 21.](#)
- Les références aux paramètres de session, de flux et de super noeud, comme `'$P-Maxvalue'`, doivent être placées entre guillemets simples.
- Si des guillemets doubles sont utilisés, une expression est traitée comme un littéral de chaîne—par exemple, `"Web graph of BP and Drug"`. Cela peut engendrer des résultats inattendus si des guillemets simples et doubles ne sont pas utilisés avec précaution. Par exemple, `"$P-Maxvalue"` aura plutôt l'air d'une chaîne que d'une référence à la valeur stockée dans un paramètre.
- Les noms de fichier, tels que `"druglearn.str"`, doivent être placés entre guillemets doubles.
- Les noms de noeud, comme `databaseNode` ou `Na_to_K` peuvent être utilisés sans guillemets ou entre guillemets simples. *Remarque* : les noms doivent être placés entre guillemets s'ils contiennent des espaces ou des caractères spéciaux. Toutefois, vous ne pouvez pas utiliser un nom de noeud dans un script si ce nom commence par un chiffre (comme `'2a_referrerID'`).
- Les propriétés booléennes doivent être lues ou définies à l'aide des valeurs `true` et `false` (faux), saisies tout en minuscules. Les variations qui contiennent `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `False`, `FALSE`, ou `0` sont également reconnues lors de la définition des valeurs mais peuvent parfois générer des erreurs lors de la lecture des valeurs de propriété. Toute autre

valeur sera interprétée comme étant true (vrai). Utiliser true et false en minuscule de façon cohérente permet d'éviter toute confusion.

- Les chaînes littérales ou les blocs contenant des sauts de ligne, des espaces, des guillemets simples ou doubles peuvent être placés entre guillemets triples. [Pour plus d'informations, reportez-vous à la section Blocs de texte littéral sur p. 30.](#)
- Les expressions CLEM, telles que "Age >= 55", doivent être placées entre guillemets doubles, comme l'illustre l'exemple suivant :


```
set :derivenode.flag_expr = "Age >= 55"
```
- Si vous utilisez des guillemets à l'intérieur d'une expression CLEM, veillez à faire précéder chaque guillemet d'une barre oblique inverse (\), par exemple :


```
set :node.parameter = "BP = \"HIGH\""
```

Même si ces directives ne sont pas obligatoires dans toutes les instances, il est recommandé de les appliquer, et ce pour plus de clarté. Le vérificateur de scripts, disponible dans toutes les boîtes de dialogue de génération de scripts, signale les syntaxes ambiguës.

Référence aux noeuds

Vous pouvez faire référence aux noeuds dans des scripts de différentes manières :

- Vous pouvez attribuer un nom aux noeuds, par exemple DRUG1n. Un nom peut être qualifié par son type ; par exemple, Drug:neuralnetworknode fait référence au noeud de réseau de neurones nommé Drug uniquement et à aucun autre type de noeud.
- Vous pouvez spécifier des noeuds par type uniquement. Par exemple, :neuralnetworknode fait référence à tous les noeuds R. neurones (Réseau de neurones). Tous les types de noeud valides peuvent être utilisés—par exemple, samplenode, neuralnetworknode, et kmeansnode. Le suffixe node est facultatif et peut être omis ; il est toutefois recommandé de l'ajouter car il facilite l'identification des erreurs dans les scripts.
- Vous pouvez référencer chaque noeud par son ID unique, comme affiché dans l'onglet Annotations de chaque noeud. Utilisez un symbole "@" suivi de l'ID ; par exemple, @id5E5GJK23L.custom_name = "My Node". [Pour plus d'informations, reportez-vous à la section Annotations dans le chapitre 5 dans *Guide de l'utilisateur de IBM SPSS Modeler 15.*](#)

Modèles générés. Les mêmes règles s'appliquent aux noeuds de modèle généré. Vous pouvez utiliser le nom de noeud tel qu'il apparaît dans la palette de modèles générés de la fenêtre des questionnaires, ou faire référence aux noeuds de modèle généré en fonction du type. Les noms utilisés pour faire référence aux modèles générés figurant dans le questionnaire sont différents de ceux utilisés pour les modèles qui ont été ajoutés à un flux à des fins de scoring (ces derniers utilisent un préfixe « apply »). [Pour plus d'informations, reportez-vous à la section Noms des nuggets de modèle dans le chapitre 4 sur p. 44.](#)

Référence aux noeuds à l'aide de variables

Vous pouvez utiliser des noms et des types de noeud comme valeurs de variables de script locales en utilisant le caret (^) comme syntaxe. Par exemple, quand un nom de noeud est nécessaire, ^n désigne le noeud dont le nom est stocké dans la variable n et Drug:^t fait référence au noeud Drug, dont le type est stocké dans la variable t.

Les références de noeud peuvent être stockées dans des variables de script locales (déclarées par le biais d'une instruction var), mais pas dans des paramètres de flux, de session ou de super noeud. Pour garantir la non-ambiguïté des références aux noeuds, attribuez un ID de noeud unique à une variable lorsque vous créez le noeud.

```
var x
set x = create typenode
set ^x.custom_name = "My Node"
```

- La première ligne crée une variable nommée x.
- La deuxième ligne crée un noeud Typer et stocke une référence à ce dernier dans x. Notez que x stocke une référence au noeud proprement dit, et non le nom de ce noeud.
- La troisième ligne définit la valeur de la propriété custom_name pour le noeud sur "My Node". Le caret permet d'indiquer que x représente le nom d'une variable et non d'un noeud. (En l'absence de caret, le système recherche un noeud appelé x. Il est inutile d'utiliser le caret lors de la déclaration ou de la définition de la variable, car une commande var, par exemple, ne peut avoir qu'une variable pour objet. Toutefois, sur la troisième ligne, x pourrait désigner logiquement un nom de noeud au lieu d'une variable. Par conséquent, le caret est nécessaire pour différencier ces deux éléments.)

Les utilisateurs font souvent l'erreur de stocker une référence à un noeud dans une variable sans avoir déclaré cette dernière au préalable.

```
set x = create typenode
set ^x.custom_name = "My Node"
```

Dans ce cas, la commande SET tente de créer x en tant que paramètre de flux, de session ou de super noeud, plutôt que comme variable. En outre, elle renvoie une erreur car il est impossible de stocker une référence à un noeud dans un paramètre.

Référencement des noeuds par ID

Vous pouvez également stocker un ID de noeud unique dans une variable. Par exemple :

```
var n
set n = "id5E5GJK23L"
set @^n.custom_name = "My Node"
```

Bouclage dans les noeuds d'un flux. Vous pouvez également utiliser la propriété stream.nodes pour obtenir la liste de tous les noeuds d'un flux, puis itérer via cette liste pour accéder à chaque noeud. [Pour plus d'informations, reportez-vous à la section Rapport de flux dans le chapitre 6 sur p. 68.](#)

Exemples

NAME:TYPE

NAME est le nom du noeud, et TYPE indique son type. Vous devez au moins indiquer NAME ou TYPE. Vous pouvez omettre l'un de ces éléments, mais pas les deux. Par exemple, la commande suivante crée un noeud Calculer entre deux noeuds existants : le noeud Délimité drug1n et un noeud Nuage (le caractère deux-points n'est pas utilisé dans les nouveaux noeuds) :

```
create derivenode connected between drug1n and :plotnode
```

Vous pouvez également faire précéder NAME ou TYPE d'un caret (^) pour indiquer la présence d'un paramètre ; par exemple :

Drug:^t

Cette référence désigne un noeud Drug, où t est un paramètre qui définit le type de noeud. Par exemple, si la valeur de ^t est c50node, la référence précédente peut être traduite sous la forme suivante :

Drug:c50node

De la même manière, il est possible d'utiliser un paramètre pour le nom du noeud. Par exemple, il est possible d'utiliser les références suivantes dans un contexte où un nom de noeud est requis :

^n:derivenode
^n

Extraction d'objets

La commande get renvoie une référence à un objet de flux, de noeud ou de sortie, ce qui permet de manipuler ces objets à l'aide de scripts. Par exemple :

```
var mynode
set mynode = get node flag1:derivenode
position ^mynode at 400 400

var mytable = get output :tableoutput
export output ^mytable as c:/mytable.htm format html

set stream = get stream 'Stream1'
set ^stream.execute_method = "Script"
```

Définition de l'objet actuel

Vous pouvez utiliser les variables spéciales suivantes pour faire référence aux objets actuels :

- node
- stream
- output
- project

A l'exception de la variable `project`, elles peuvent toutes être également réinitialisées de façon à modifier le contexte actuel. Contrairement aux autres variables de script, il est inutile de les déclarer d'abord à l'aide de la commande `var` puisqu'elles sont prédéfinies.

```
set node = create typenode
rename ^node as "mytypenode"

set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

Etant donné que ces variables spéciales correspondent au nom des objets auxquels elles font référence, il peut être parfois difficile de différencier la variable de l'objet. [Pour plus d'informations, reportez-vous à la section Commande set dans le chapitre 4 sur p. 34.](#)

Commentaires

L'affectation d'une valeur d'un type incorrect à une variable spéciale (tel que le paramétrage d'un objet de noeud sur la variable `stream`) provoque une erreur d'exécution.

Partout où la variable spéciale peut être utilisée, toute autre variable peut l'être également. Par exemple, vous pouvez enregistrer le flux actuel à l'aide de l'instruction suivante :

```
save stream as 'C:/My Streams/Churn.str'
```

La ligne suivante est également valide :

```
save my_stream as 'C:/My Streams/Churn.str'
```

Dans cette instruction, vous avez préalablement affecté à `my_stream` la valeur `stream`.

Ouverture de flux et d'autres objets

Dans un script autonome, vous pouvez ouvrir un flux en indiquant le nom et l'emplacement du fichier. Par exemple :

```
open stream "c:/demos/druglearn.str"
```

Il est possible d'ouvrir d'autres types d'objet à l'aide de la commande `load`. Par exemple :

```
load node c:/mynode.nod
```

```
load model c:/mymodel.gm
```

Commande `open stream` et commande `load stream`. La commande `load stream` ajoute le flux indiqué à l'espace de travail, et ce sans effacer les noeuds du flux en cours. Cette commande, très fréquemment utilisée dans les versions précédentes de Clementine, a été largement supplantée dans les nouvelles versions par la possibilité d'ouvrir, de gérer et de copier des noeuds entre plusieurs flux.

Utilisation de plusieurs flux

Mis à part les commandes utilisées pour accéder aux flux à partir du système de fichiers ou du IBM® SPSS® Collaboration and Deployment Services Repository (*open*, *load*, et *retrieve*), la plupart des commandes de script s'appliquent automatiquement au flux actuel. Dans les scripts autonomes, toutefois, il se peut que vous souhaitiez ouvrir et manipuler plusieurs flux à partir du même script. Pour cela, définissez une référence à un flux ouvert ou utilisez la commande *with... endwith* pour réaffecter temporairement le flux actuel.

Par exemple, pour fermer un flux autre que le flux actuel, vous pouvez utiliser la commande *get stream* pour faire référence au flux souhaité :

```
set stream = get stream "druglearn"  
close stream
```

Ce script réaffecte la variable spéciale « stream » au flux *druglearn* (celui-ci devient alors le flux actuel), puis ferme le flux.

Il est également possible de réaffecter temporairement le flux actuel à l'aide de l'instruction *with stream*. Par exemple :

```
with stream 'druglearn'  
  create typenode  
  execute_script  
endwith
```

Les instructions ci-dessus exécutent l'action *create*, ainsi que le script du flux, en utilisant comme flux actuel le flux spécifié. Le flux actuel d'origine est restauré une fois que chaque instruction a été exécutée. Vous pouvez aussi inclure des instructions conditionnelles et des éléments de boucle, comme l'illustre l'exemple suivant :

```
with stream 'druglearn'  
  create tablenode at 500 400  
  create selectnode connected between :typenode and :tablenode  
  for l from 1 to 5  
    set :selectnode.condition = 'Age > ' >< (l * 10)  
    execute :selectnode  
  endfor  
endwith
```

Cette instruction définit **STREAM** comme flux actuel pour toutes les expressions comprises dans la boucle et restaure la valeur d'origine quand la boucle est terminée.

Variabes de script locales

Les variables de script locales sont déclarées à l'aide de la commande *var* et définies pour le script actuel uniquement. Les variables sont différentes des paramètres ; en effet, ces derniers peuvent être définis pour une session, un flux ou un super noeud, et ne peuvent contenir que des chaînes ou des valeurs numériques.

```
var my_node
set my_node = create distributionnode
rename ^my_node as "Distribution of Flag"
```

Lorsque vous faites référence à des variables existantes, veillez à utiliser le caret (^) devant le nom du paramètre. Par exemple, si l'on considère le script précédent :

- La première ligne déclare la variable.
- La deuxième ligne définit la valeur correspondante.
- La troisième ligne, quant à elle, renomme le noeud référencé par la variable (et non la variable proprement dite). Le caret signifie que ^my_node (^mon_noeud) représente le nom d'une variable au lieu du nom littéral du noeud. (En l'absence de caret, la commande rename recherche un noeud appelé my_node. Le caret n'est pas nécessaire dans la première ou la deuxième ligne car l'objet d'une commande var ne peut être qu'une variable. Le caret n'est utilisé que pour faire référence à une variable déjà définie, auquel cas le fait de le supprimer générerait une référence ambiguë.)
- Lors de la résolution des références aux variables, le programme vérifie la liste des variables locales avant celle des paramètres de session, de flux ou de super noeud. Par exemple, si la variable x existe à la fois comme variable locale et comme paramètre de session, l'utilisation de la syntaxe '\$P-X' dans une instruction de script garantit l'utilisation du paramètre de session au lieu de la variable locale.

Remarque : Concrètement, si vous définissez une variable sans la déclarer au préalable à l'aide de la commande var, un paramètre de flux, de session ou de super noeud est créé (en fonction du contexte du script actuel). Par exemple, le code suivant crée une variable de script locale nommée z et lui affecte la valeur [1 2 3] :

```
var z
set z = [1 2 3]
```

En cas d'omission de la commande var (et en admettant qu'aucune variable ou qu'aucun noeud du même nom n'existe déjà), z est créé en tant que paramètre et non en tant que variable.

Paramètres de flux, de session et de super noeud

Vous pouvez définir des paramètres afin de les utiliser dans les expressions CLEM et pour la génération de scripts. Il s'agit, en effet, de variables définies par l'utilisateur qui sont enregistrées et conservées dans le flux, la session ou le super noeud en cours ; elles sont accessibles à partir de l'interface utilisateur, ainsi que via la fonction de génération de scripts. Si vous enregistrez un flux, par exemple, les paramètres définis pour ce dernier sont également enregistrés. (Cela permet de les différencier des variables de script locales qui ne peuvent être utilisées que dans le script dans lequel elles sont déclarées.) Dans le cadre de la génération de scripts, les paramètres sont souvent intégrés dans une expression CLEM et la valeur du paramètre est spécifiée dans le script.

La portée d'un paramètre dépend de l'endroit où ce dernier est défini :

- Les paramètres de flux peuvent être définis dans un script de flux ou dans la boîte de dialogue des propriétés du flux ; ils sont disponibles pour tous les noeuds du flux. Ils sont affichés sur la liste des paramètres dans le Générateur de formules.
- Les paramètres de session peuvent être définis dans un script autonome ou dans la boîte de dialogue des paramètres de session. Ils sont disponibles pour tous les flux utilisés dans la session actuelle (tous les flux répertoriés dans l'onglet Flux du panneau des gestionnaires).

Il est également possible de définir des paramètres pour les super noeuds ; dans ce cas, les paramètres ne sont visibles que par les noeuds encapsulés au sein de ce super noeud. [Pour plus d'informations, reportez-vous à la section Définitions des paramètres de super noeud dans le chapitre 9 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Définition de paramètres dans les scripts

Vous pouvez définir les paramètres des scripts à l'aide de la commande `set` et de la syntaxe suivante :

```
set foodtype = pizza
```

Si aucune variable ni aucun noeud nommé `foodtype` n'est déclaré dans le script actuel, cette commande crée le paramètre `foodtype` dont la valeur par défaut est `pizza`.

Interface utilisateur. Vous pouvez également définir ou afficher les paramètres via l'interface utilisateur en sélectionnant Propriétés du flux ou Définir les paramètres de session dans le menu Outils. Ces boîtes de dialogue vous permettent, en outre, d'indiquer des options supplémentaires (telles que le type de stockage) qui ne sont pas disponibles par le biais de la génération de scripts. [Pour plus d'informations, reportez-vous à la section Définition des paramètres de flux et de session dans le chapitre 5 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Ligne de commande. Vous pouvez également définir les paramètres à partir de la ligne de commande, auquel cas ils sont créés en tant que paramètres de session.

Référence à des paramètres dans les scripts

Vous pouvez faire référence aux paramètres précédemment créés en les encapsulant entre des guillemets simples et en les faisant précéder de la chaîne `$P`, par exemple, '`$P-minvalue`'. Vous pouvez également utiliser uniquement le nom du paramètre, comme `minvalue`. La valeur d'un paramètre est toujours une chaîne ou un nombre. Par exemple, vous pouvez faire référence au paramètre `foodtype` et définir une nouvelle valeur à l'aide de la syntaxe suivante :

```
set foodtype = pasta
```

Vous pouvez également faire référence aux paramètres se trouvant dans le contexte d'une expression CLEM utilisée dans un script. Le script suivant est un exemple. Il définit les propriétés d'un noeud Sélectionner afin d'inclure les enregistrements dans lesquels la valeur `Age` est

supérieure à celle spécifiée par le paramètre de flux cutoff (limite). Le paramètre est utilisé dans une expression CLEM avec la syntaxe appropriée de CLEM—'\$P-cutoff' :

```
set :selectnode {
mode = "Include"
condition = "Age >= '$P-cutoff'"
}
```

Le script ci-dessus utilise la valeur par défaut du paramètre de flux cutoff (limite). Vous pouvez spécifier une nouvelle valeur de paramètre en ajoutant la syntaxe suivante au-dessus des spécifications du noeud Sélectionner :

```
set cutoff = 50
```

Le script obtenu sélectionne tous les enregistrements dans lesquels la valeur Age est supérieure à 50.

Pour plus d'informations, reportez-vous à la section Paramètres de flux, de session et de super noeud dans le chapitre 7 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.

Contrôle de l'exécution du script

L'exécution des scripts traite généralement les instructions une par une. Cependant, vous pouvez passer outre à ce mode d'exécution, en utilisant une instruction conditionnelle if (si) et diverses boucles for. Par exemple :

```
if s.maxsize > 10000 then
s.maxsize = 10000
connect s to :derive
endif
```

Les boucles for se présentent sous diverses formes :

```
for PARAMETER in LIST
STATEMENTS
endfor
```

Ce script exécute les instructions STATEMENTS une seule fois pour chaque valeur de la liste LIST attribuée au paramètre PARAMETER, dans l'ordre de la liste. La liste n'est pas insérée entre crochets et elle est composée de constantes. Plusieurs autres formes sont également disponibles. Pour plus d'informations, reportez-vous à la section Commandes de script générales dans le chapitre 4 sur p. 32.

Opérateurs de scripts

En plus des opérateurs CLEM habituels, vous pouvez manipuler les variables de script locales (déclarées à l'aide d'une commande var) à l'aide des opérateurs "+" et "-". L'opérateur « + » ajoute un élément à la liste alors que l'opérateur « - » en supprime. Voici un exemple :

```
var z # créez une nouvelle variable locale
set z = [1 2 3] # définissez-la dans la liste contenant 1, 2, et 3
```

```
set z = z + 4 # ajoutez un élément; z est maintenant égal à [1 2 3 4]
```

Vous ne pouvez pas utiliser ces opérateurs avec des paramètres de flux, de super noeud ou de session (définis dans les scripts à l'aide de la commande `set`), ni en dehors des scripts figurant dans les expressions CLEM générales (par exemple, une formule dans un noeud Calculer).

Expressions CLEM dans les scripts

Vous pouvez utiliser des expressions, des fonctions et des opérateurs CLEM dans les scripts IBM® SPSS® Modeler. Cependant, votre expression de script ne peut pas contenir d'appel à des fonctions `@`, à des fonctions de date/heure ou à des opérations sur les bits. De plus, les règles suivantes s'appliquent aux expressions CLEM utilisées pour la génération de scripts :

- Les paramètres doivent être placés entre guillemets simples et précédés du préfixe `$P-`.
- Les expressions CLEM doivent être placées entre guillemets. Si l'expression CLEM contient elle-même des chaînes ou des noms de champs entre guillemets, ces guillemets doivent être précédés d'une barre oblique inverse (`\`). [Pour plus d'informations, reportez-vous à la section Syntaxe pour les scripts sur p. 20.](#)

Dans le cadre de la génération de scripts, vous pouvez utiliser des valeurs globales comme `GLOBAL_MEAN(Age)`, mais vous ne pouvez pas utiliser la fonction `@GLOBAL` à l'intérieur de l'environnement de génération de scripts.

Voici des exemples d'expressions CLEM utilisées pour générer des scripts :

```
set :balancenode.directives = {{1.3 "Age > 60"}}
set :fillernode.condition = "(Age > 60) and (BP = \"High\")"
set :derivernode.formula_expr = "substring(5, 1, Drug)"
set Flag:derivernode.flag_expr = "Drug = X"
set :selectnode.condition = "Age >= '$P-cutoff'"
set :derivernode.formula_expr = "Age - GLOBAL_MEAN(Age)"
```

Insertion de commentaires et de lignes incomplètes

Les caractères suivants sont utilisés dans la génération de scripts pour signaler les commentaires et les lignes incomplètes :

Caractère	Utilisation	Exemple
#	Le signe dièse signale un commentaire. La suite de la ligne est ignorée.	#Ceci est un commentaire d'une seule ligne.
\	Une ligne terminée par une barre oblique inverse indique que l'instruction continue sur la ligne suivante.	Reportez-vous à l'exemple ci-dessous.

Caractère	Utilisation	Exemple
/*	La séquence /* indique le début d'un commentaire. Toutes les instructions sont ignorées, jusqu'à ce que le programme trouve le marqueur de fin de commentaire */.	Reportez-vous à l'exemple ci-dessous.
"""	Les chaînes littérales ou les blocs contenant des sauts de ligne, des espaces, des guillemets simples ou doubles peuvent être placés entre guillemets triples. Pour plus d'informations, reportez-vous à la section Blocs de texte littéral sur p. 30.	

Exemples

```
/* This is a
multi line
comment
*/
```

```
#voici une instruction de plusieurs lignes
set :fixedfilenode.fields = [{"Age" 1 3}\
{"Sex" 5 7} {"TA" 9 10} {"Cholesterol" 12 22}\
{"Na" 24 25} {"K" 27 27} {"Drug" 29 32}]
```

Blocs de texte littéral

Vous pouvez mettre des blocs de texte littéral dotés d'espaces, de tabulations et de sauts de ligne entre guillemets triples pour les inclure dans un script. Le texte du bloc entre guillemets apparaît en tant que texte littéral, y compris les espaces, les sauts de ligne, ainsi que les guillemets simples et doubles imbriqués. Aucun caractère d'échappement ou de continuation de ligne n'est nécessaire.

Par exemple, cette méthode peut servir à imbriquer un ensemble de directives d'accroissement d'arbre dans un script comme suit :

```
set :cartnode.tree_directives = """
Create Root_Node
Grow Node Index 0 Children 1 2 SplitOn ("DRUG",
  Group ("drugA", "drugB", "drugC" )
  Group ("drugY", "drugX" ))
End Tree
"""
```

Cette méthode s'avère également utile pour les chemins et les annotations, par exemple :

```
set :node.annotation = """This node was built to help identify which of the following indicators
Dairy
Fish
Vegetable
Meat
Pastries
Confectionary
is showing unusual sales behaviour"""
```

IBM® SPSS® Modeler ignore les sauts de ligne s'ils sont placés après le marqueur de littéral d'ouverture. Voici un exemple identique à l'exemple précédent :

```
set :node.annotation = ""
```

```
Ce noeud a été conçu pour identifier lequel des indicateurs utilisés
```

```
Etc...
```

```
""
```

Commandes de script

Cette section récapitule les commandes pouvant être utilisées dans les scripts IBM® SPSS® Modeler, organisées par type d'objet. Pour plus d'informations sur le langage de script, reportez-vous à [le chapitre 3](#). Pour plus d'informations sur les propriétés de noeud, flux, projet et super noeud, reportez-vous à le chapitre 9 dans le chapitre 22.

Commandes de script générales

Sauf indication contraire, les commandes suivantes sont disponibles dans tous les scripts autonomes, ainsi que dans les scripts de flux et de super noeud.

execute_all

```
execute_all
```

Exécute tous les noeuds terminaux du flux actuel.

```
open stream "c:/demos/druglearn.str"  
execute_all
```

execute_script

```
execute_script
```

Scripts autonomes uniquement. Exécute le script de flux associé au flux actuel. (commande réservée aux seuls scripts autonomes puisque, sinon, le script de flux s'appellerait lui-même.)

```
open stream "c:/demos/mysample.str"  
execute_script
```

exit

```
exit CODE
```

Quitte le script en cours. Le code de sortie peut permettre d'évaluer le script ou la condition d'un flux ou d'un noeud ; par exemple :

```
create tablenode  
create variablefilenode  
connect :variablefilenode to :tablenode
```

```
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"  
execute 'Table'
```

```
set param = value :tablenode.output at 1 1
```

```

if ^param = 23 then
  create derivenode
else exit 2
endif

```

for...endfor

La commande for...endfor parcourt un ensemble d'instructions en boucle en fonction d'une condition. Cette commande peut revêtir différentes formes, obéissant toutes à la même structure globale.

```

for PARAMETER in LIST
  STATEMENTS
endfor

```

for PARAMETER in LIST. Exécute STATEMENTS une seule fois pour chaque valeur de LIST attribuée au PARAMETER, dans l'ordre de la liste. Par exemple, la propriété Filter.include peut être définie sur true pour les champs multiples, comme suit :

```

for f in Age Sex
  set Filter.include.^f=true
endfor

```

for PARAMETER from N to M. Exécute les STATEMENTS une seule fois pour chaque entier compris entre N et M inclus ; par exemple :

```

for I from 1 to 5
  set :selectnode.condition = 'Age > ' >< (I * 10)
  execute :selectnode
endfor

```

for PARAMETER in_fields_to NODE. Exécute les STATEMENTS une seule fois pour chaque champ disponible en amont du NODE. Par exemple, la commande suivante permet d'affecter la propriété include sur true pour tous les champs (y compris ceux précédemment paramétrés sur false :

```

for f in_fields_to Filter
  set Filter.include.^f = "true"
endfor

```

Remarque : Lorsqu'un noeud peut avoir plusieurs champs d'entrée avec le même nom (par exemple Fusionner ou Ajouter), cette méthode renvoie la liste des champs situés en aval plutôt qu'en amont, pour éviter tout conflit qui pourrait en résulter.

for PARAMETER in_fields_at NODE. Exécute les STATEMENTS une seule fois pour chaque champ provenant du (ou en aval du) NODE spécifié. Ainsi, si le noeud est un noeud Filtrer, seuls les champs qui passent à travers sont inclus, et le noeud ne doit pas être un noeud terminal car aucun champ ne serait renvoyé. Par exemple, contrairement à l'exemple ci-dessus, le script suivant n'a aucun effet car la boucle ne s'exécute que sur les champs déjà paramétrés sur la valeur true :

```

for f in_fields_at Filter
  set Filter.include.^f = "true"

```

```
endfor
```

for PARAMETER in_models. Exécute les STATEMENTS une seule fois pour chaque nugget de modèle figurant dans la palette Modèles. Par exemple, le script suivant permet d'insérer dans le flux actuel chaque modèle de la palette. (La variable xpos évite d'empiler les noeuds l'un au-dessus de l'autre dans l'espace de travail de flux.)

```
var xpos
set xpos = 100
for m in_models
  set xpos = xpos + 100
  insert model ^m at ^xpos 100
endifor
```

for PARAMETER in_streams. *Scripts autonomes uniquement.* Exécute les STATEMENTS une seule fois pour chaque flux chargé (selon l'ordre dans lequel les flux apparaissent dans la palette de flux). Si PARAMETER est la variable spéciale stream, le flux actuel est utilisé pour les STATEMENTS dans la boucle. La valeur d'origine de stream est restaurée quand la boucle prend fin.

if...then...else...

```
if EXPR then
  STATEMENTS 1
else
  STATEMENTS 2
endif
```

Exécute STATEMENTS 1 si l'expression spécifiée est vraie et STATEMENTS 2 si l'expression est fausse. La clause else est facultative.

```
if :sampler.use_max_size = true then
  set x = "yes"
else
  set x = "no"
endif
```

Commande set

```
set VARIABLE = EXPRESSION
set PARAMETER = EXPRESSION
set PROPERTY = EXPRESSION
```

Définit la valeur d'une variable de script locale, d'une variable spéciale, d'un paramètre ou d'une propriété.

Définition de variables

Pour définir la valeur d'une variable de script locale, vous devez d'abord déclarer la variable à l'aide de la commande `var` ; par exemple :

```
var xpos
var ypos
set xpos = 100
set ypos = 100
```

La valeur de la variable peut être une expression CLEM valide pour la génération de scripts, une commande de script qui renvoie une valeur (telle que `load`, `create` ou `get`) ou bien encore une valeur littérale.

```
set xpos = ^xpos + 50
```

```
var x
set x = create typenode
```

```
var s
set s = get stream 'Druglearn'
```

Définition de variables spéciales pour des objets de référence

Les variables spéciales `node`, `stream`, `output` et `project` permettent de faire référence à l'objet en cours dans chaque contexte. A l'exception de la variable `project`, elles peuvent toutes être également réinitialisées de façon à modifier le contexte actuel. Contrairement aux autres variables de script, il est inutile de les déclarer d'abord à l'aide de la commande `var` puisqu'elles sont prédéfinies.

```
set node = create typenode
rename ^node as "mytypenode"
```

```
set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

Très utiles, ces variables présentent de subtiles différences dans leur utilisation, comme l'illustre l'exemple suivant :

```
set stream = get stream 'Stream7'
set ^stream.execute_method = "Script"
save stream as c:/sample7.str
close stream
```

- La première ligne réinitialise le flux en cours ou, plus précisément, définit la valeur de la variable spéciale `stream`. (Autrement dit, `stream` est une variable et non une partie de la commande.)

- La deuxième ligne utilise cette variable pour définir une propriété du flux en cours (pour plus d'informations sur les propriétés, voir ci-dessous). Le caret permet d'indiquer que `^stream` est le nom d'une variable et non d'un objet tel qu'un noeud. (En l'absence de caret, la commande `set` recherche un noeud appelé `stream`.)
- Les deux dernières lignes enregistrent et ferment le flux en cours. `stream` reste, comme avant, une variable, mais aucun caret ne peut être utilisé dans ce cas car les commandes `save` et `close`, telles qu'elles sont utilisées dans notre exemple, ne peuvent s'appliquer qu'à un flux. (Le caret n'est généralement utilisé que lorsque sa présence est indispensable pour lever une ambiguïté de référence.)

Référencement du projet en cours. Vous pouvez utiliser la variable spéciale `project` pour faire référence au projet en cours (voir l'exemple de définition des propriétés d'un projet, ci-dessous). Il est impossible de réinitialiser la valeur de `project` car un seul projet peut être ouvert à la fois (il n'existe donc qu'un projet en cours à chaque fois).

Définition de paramètres

Vous pouvez définir les paramètres de flux, de session et de super noeud de la même manière que les variables mais sans utiliser la commande `var`.

```
set p = 1
set minvalue = 21
```

Remarque : En pratique, si l'objet d'une commande `set` ne correspond pas au nom d'une variable déclarée, d'une variable spéciale ou d'un objet existant (tel qu'un noeud), un paramètre est créé. [Pour plus d'informations, reportez-vous à la section Paramètres de flux, de session et de super noeud dans le chapitre 3 sur p. 26.](#)

Définition des propriétés d'un noeud, d'un flux et d'un projet

Vous pouvez également définir les propriétés de noeuds, de flux et de projets, par exemple :

```
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"

set ^stream.execute_method = "Script"

load project "C:/myproject.cpj"
set ^project.structure = Phase
```

Pour obtenir la liste complète des propriétés disponibles pour les noeuds, les flux et les projets, reportez-vous à [Référence sur les propriétés sur p. 115](#).

Définition de plusieurs propriétés. Vous pouvez attribuer plusieurs expressions aux propriétés des noeuds ou d'autres objets en une seule opération. Utilisez cette méthode quand vous devez apporter plusieurs modifications à un noeud avant que le modèle de données soit défini. Le format permettant de définir des propriétés multiples est le suivant :

```
set NODE {
  NODEPROPERTY1 = EXPRESSION1
  NODEPROPERTY2 = EXPRESSION2
}
```

Par exemple :

```
set :samplene {
  max_size = 200
  mode = "Include"
  sample_type = "First"
}

set ^project {
  summary = "La modélisation initiale fonctionne sur les dernières données"
  ordering = NameAddedType
}
```

Définition de valeurs booléennes (true (vrai) et false (faux)). Lors de la lecture ou de l'écriture de propriétés de type Booléen, les valeurs true et false doivent être en minuscules, par exemple :

```
set :variablefilenode.read_field_names = true
```

Remarque : Les variations qui contiennent Off, OFF, off, No, NO, no, n, N, f, F, false, False, FALSE, ou 0 sont également reconnues lors de la définition des valeurs mais peuvent parfois générer des erreurs lors de la lecture des valeurs de propriété. Toute autre valeur sera interprétée comme étant True (vrai). Utiliser true et false en minuscule de façon cohérente permet d'éviter toute confusion.

Exemple : Définition de propriétés de noeud

De nombreuses propriétés propres aux noeuds (appelées parfois propriétés de paramètre) permettent de définir les options des boîtes de dialogue de l'interface utilisateur pour les différents noeuds. Par exemple, pour créer un flux et spécifier les options de chaque noeud, vous pouvez utiliser un script semblable à celui indiqué ici. Pour plus d'informations sur les propriétés de noeud, flux, projet et super noeud, reportez-vous à le chapitre 9 dans le chapitre 22.

```
create varfilenode at 100 100
set :varfilenode {
  full_filename = "demos/drug1n"
  read_field_names = true
}

create tablenode at 400 100
create samplene connected between :varfilenode and :tablenode
set :samplene {
  max_size = 200
  mode = "Include"
  sample_type = "First"
}

create plotnode at 300 300
create derivene connected between drug1n and :plotnode
set :derivene {
  new_name = "Ratio of Na to K"
  formula_expr = "'Na' / 'K'"
}

set :plotnode {
  x_field = 'Ratio of Na to K'
  y_field = 'Age'
```

```
color_field = 'BP'
}
```

Commande var

```
var VARNAME
```

Déclare une variable de script locale.

```
var my_node
set my_node = create distributionnode
rename ^my_node as "Distribution of Flag"
```

Les variables sont différentes des paramètres ; en effet, ces derniers peuvent être définis pour une session, un flux ou un super noeud, et ne peuvent contenir que des chaînes ou des valeurs numériques. Concrètement, si vous définissez une variable sans la déclarer au préalable à l'aide de la commande VAR, un paramètre de flux, de session ou de super noeud est créé (en fonction du contexte du script actuel). [Pour plus d'informations, reportez-vous à la section Variables de script locales dans le chapitre 3 sur p. 25.](#)

Objets de noeud

Les commandes de script disponibles pour les objets de noeud sont les suivantes.

create NODE

```
create NODE
create NODE at X Y
create NODE between NODE1 and NODE2
create NODE connected between NODE1 and NODE2
```

Crée un noeud du type indiqué, par exemple :

```
create statisticsimportnode
```

Vous pouvez également, si vous le souhaitez, indiquer une position et des options de connexion :

```
create featureselectionnode at 400 100
```

```
create typenode between :statisticsimportnode and :featureselectionnode
```

```
create selectnode connected between :typenode and :featureselectionnode
```

Pour éviter toute ambiguïté, vous pouvez également créer un noeud à l'aide de variables. Dans l'exemple ci-dessous, un noeud Typenode est créé et la variable de référence *x* est définie pour contenir une référence à ce noeud Typenode. Vous pouvez ensuite utiliser la variable *x* pour renvoyer l'objet référencé par *x* (dans ce cas, le noeud Typenode) et effectuer d'autres opérations (par exemple, renommer, positionner ou connecter le nouveau noeud).

```
var x
set x = create typenode
```

```

rename ^x as "mytypenode"
position ^x at 200 200
var y
set y = create varfilenode
rename ^y as "mydatasource"
position ^y at 100 200
connect ^y to ^x

```

Dans l'exemple ci-dessus, deux noeuds sont créés, renommés, positionnés, puis connectés dans l'espace de travail.

Figure 4-1
Noeuds créés à l'aide de variables



Vous pouvez également utiliser la variable spéciale (prédéfinie) `node` de la même manière que les variables `x` et `y` mentionnées dans l'exemple ci-dessus. Il est alors inutile de déclarer la variable par le biais de la commande `var` (puisque cette variable est prédéfinie), ce qui peut rendre le script final plus lisible.

```

set node = create typenode
rename ^node as "mytypenode"
position ^node at 200 200
set node = create varfilenode
rename ^node as "mydatasource"
position ^node at 100 200
connect mydatasource to mytypenode

```

Remarque : Vous pouvez réutiliser les variables spéciales, telles que `node`, pour faire référence à plusieurs noeuds. Utilisez simplement la commande `set` pour réinitialiser l'objet référencé par la variable. [Pour plus d'informations, reportez-vous à la section Définition de l'objet actuel dans le chapitre 3 sur p. 23.](#)

Duplication de noeuds. Vous pouvez également utiliser la commande `duplicate` pour dupliquer un noeud existant. [Pour plus d'informations, reportez-vous à la section duplicate NODE sur p. 40.](#)

connect NODE

```

connect NODE1 to NODE2
connect NODE1 between NODE2 and NODE3

```

Connecte le noeud `NODE1` aux autres noeuds indiqués.

```
connect :statisticsimportnode to :typenode
```

```
connect :selectnode between :typenode and :featureselectionnode
```

delete NODE

```
delete NODE
```

Supprime du flux actuel le noeud spécifié.

```
delete :statisticsimportnode
```

```
delete DRUG1N:variablefilenode
```

disable NODE

```
disable NODE
```

Désactive le noeud spécifié du flux en cours. Le noeud est ignoré durant l'exécution du flux. Cela vous évite d'avoir à supprimer ou à contourner le noeud et signifie que vous pouvez le laisser connecté aux noeuds restants. Vous pouvez toujours modifier les paramètres du noeud, cependant les modifications ne prennent pas effet tant que vous n'avez pas de nouveau activé le noeud.

```
disable :statisticsimportnode
```

```
disable DRUG1N:variablefilenode
```

disconnect NODE

```
disconnect NODE
```

```
disconnect NODE1 from NODE2
```

```
disconnect NODE1 between NODE2 and NODE3
```

Déconnecte le noeud indiqué de tous les autres noeuds (action par défaut) ou des noeuds définis.

```
disconnect :typenode
```

```
disconnect :typenode from :selectnode
```

duplicate NODE

```
duplicate NODE as NEWNAME
```

Crée un double du noeud indiqué. Vous pouvez éventuellement indiquer la position de manière absolue ou relative.

```
duplicate :derivenode as flag1 at 100 400
```

```
duplicate flag1 as flag2 connected between flag1 and flag3
```

enable NODE

```
enable NODE
```

Active un noeud précédemment désactivé dans le flux en cours. Le noeud est inclus durant l'exécution du flux. Si vous avez modifié les paramètres du noeud alors qu'il était désactivé, les modifications prennent effet maintenant.

```
enable :statisticsimportnode
```

```
enable DRUG1N:variablefilenode
```

execute NODE

```
execute NODE
```

Exécute le noeud indiqué, par exemple :

```
execute :neuralnetworknode
```

Dans le cas d'un noeud non terminal, l'opération exécutée correspond à l'option Exécuter à partir d'ici du menu déroulant.

Pour exécuter tous les noeuds terminaux du flux actuel, entrez la commande suivante :

```
execute_all
```

Scripts autonomes uniquement. Pour exécuter le script de flux associé au flux actuel :

```
execute_script
```

Remarque : vous exécutez les scripts associés à différents flux en désignant un flux comme flux actuel ou en utilisant la commande with. [Pour plus d'informations, reportez-vous à la section Utilisation de plusieurs flux dans le chapitre 3 sur p. 25.](#)

export NODE as FILE

```
export node NODE in DIRECTORY format FORMAT  
export node NODE as FILE format FORMAT
```

Exportation en PMML. Pour exporter un modèle généré au format PMML, exécutez la commande suivante :

```
export Drug as c:/mymodel.txt format pmml
```

Export SQL. Pour exporter un modèle généré au format SQL, exécutez la commande suivante :

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

Détails du noeud. Pour exporter les détails du noeud au format HTML ou texte, exécutez la commande suivante :

```
export Drug as c:\mymodel.htm format html
```

```
export Drug as c:\mymodel.txt format text
```

Récapitulatif de noeud. Pour exporter le récapitulatif de noeud au format HTML ou texte, exécutez la commande suivante :

```
export Drug summary in c:/mymodels format html  
export Drug summary as c:/mymodel.txt format text  
export 'assocapriori' as 'C:/temp/assoc_apriori' format html
```

flush NODE

```
flush NODE
```

Vide le cache du noeud indiqué seulement ou de tous les noeuds du flux. Si le cache est désactivé ou n'est pas saturé pour un noeud donné, la commande n'exécute aucune opération.

```
flush :mergenode
```

Pour vider tous les noeuds du flux en cours, exécutez la commande suivante :

```
flush_all
```

get node NODE

```
get node NODE
```

Obtient une référence à un noeud existant. Vous pouvez utiliser cette commande pour vous assurer qu'aucune référence aux noeuds ambiguë n'est présente.

```
var mynode  
set mynode = get node flag1:derivenode  
position ^mynode at 400 400
```

load node FILENAME

```
load node FILENAME
```

Charge un noeud enregistré dans le flux en cours.

```
load node c:/mynode.nod
```

position NODE

```
position NODE at X Y  
position NODE between NODE1 and NODE2  
position NODE connected between NODE1 and NODE2
```

Positionne un noeud dans l'espace de travail de flux, selon un chemin absolu ou relatif. Vous pouvez également, si vous le souhaitez, indiquer des options de connexion :

```
position DRUG1n:variablefilenode at 100 100
```


position Drug:net between DRUG2n and analysis

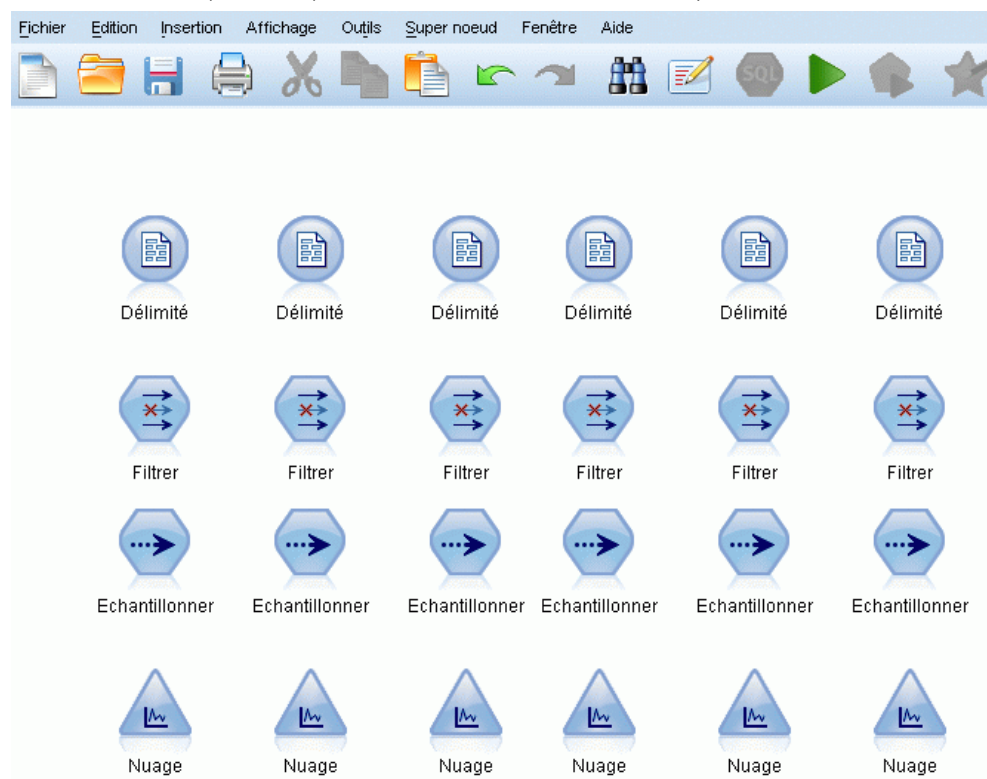
position :typenode connected between :variablefilenode and :tablenode

Positionnement des coordonnées

Pour déterminer la position des noeuds dans l'espace de travail de flux, le programme utilise une grille $x-y$ invisible. Utilisez l'illustration suivante comme référence pour les coordonnées de grille $x-y$.

Figure 4-2

Noeuds créés à la position spécifiée à l'aide des coordonnées $x-y$



rename NODE as NEWNAME

```
rename NODE as NEWNAME
```

Renomme le noeud indiqué.

```
rename :derivenode as 'Flag1'
```

```
rename :varfilenode as 'testdata'
```

retrieve node REPOSITORY_PATH

```
retrieve node REPOSITORY_PATH {label LABEL | version VERSION}
```

Récupère le noeud spécifié dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
retrieve node "/samples/drugtypenode"
```

save node NODE as FILENAME

```
save node NODE as FILENAME
```

Enregistre le noeud indiqué.

```
save node :statisticsimportnode as c:/mynode.nod
```

store node NODE as REPOSITORY_PATH

```
store node NODE as REPOSITORY_PATH {label LABEL}
```

Stocke un noeud dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
store node DRUG1n as "/samples/drug1ntypenode"
```

```
store node :typenode as "/samples/drugtypenode"
```

Objets de modèle

Les commandes de script disponibles pour les objets de modèle sont les suivantes.

Noms des nuggets de modèle

Les nuggets de modèle (également appelés modèles générés) peuvent être référencés par type, tout comme les objets de noeud et de sortie. Le tableau ci-dessous reprend les noms de référence d'objet de modèle.

Ces noms sont utilisés spécialement pour le référencement des nuggets de modèle figurant dans la palette Modèles (dans l'angle supérieur droit de la fenêtre IBM® SPSS® Modeler). Pour référencer les noeuds de modèle qui ont été ajoutés à un flux à des fins de scoring, le système utilise un ensemble différent de noms commençant par `apply...` [Pour plus d'informations, reportez-vous à la section Propriétés du noeud de nugget de modèle dans le chapitre 17 sur p. 255.](#)

Considérons l'exemple suivant : le script ajoute un nugget de modèle au flux en cours, il le connecte à un noeud Typer, puis crée et exécute un noeud Table. Comme vous pouvez le constater, le nom utilisé pour insérer le modèle de la palette est différent de celui servant à référencer le noeud de modèle « apply » une fois ce dernier ajouté au flux (`:featureselection` ou `:applyfeatureselectionnode`).

```
insert model :featureselection at 150 250
connect Type to :applyfeatureselectionnode
create tablenode at 250 250
```

```
connect :applyfeatureselectionnode to :tablenode
execute :tablenode
```

Remarque : Il ne s'agit là que d'un exemple. En règle générale, le référencement des modèles à la fois par nom *et* par type est recommandé car il permet d'éviter toute confusion (par exemple, response_01:featureselection).

Noms des nuggets de modèle (Palette Modélisation)

Nom du modèle	Modèle
anomalydetection	Anomalie
apriori	A priori
autoclassifier	Classificateur automatique
autocluster	Classification non supervisée automatique
autonumeric	Numérisation automatique
bayesnet	Réseau Bayésien
c50	C5.0
carma	Carma
arbre crt	Arbre C&RT
chaid	CHAID
coxreg	Régression de Cox
decisionlist	Liste de décision
discriminant	Analyse discriminante
facteur	ACP/Facteur
featureselection	Sélection de fonction
genlin	Régression linéaire généralisée
kmeans	K-means
knn	k plus proches voisins
kohonen	Kohonen
linéaire	Linéaire
logreg	Régression logistique
neuralnetwork	R. neurones
quest	QUEST
régression	Régression linéaire
séquence	Séquence
slrm	Modèle de réponse en auto-apprentissage
statisticsmodel	modèle IBM® SPSS® Statistics
svm	Support vector machine
timeseries	Séries temporelles
twostep	TwoStep

Noms des nuggets de modèle (Palette Modélisation de base de données)

Nom du modèle	Modèle
db2imassoc	Association ISW IBM
db2imcluster	Classification ISW IBM
db2imreg	Régression ISW IBM
db2imsequence	Séquence ISW IBM
db2imtree	Arbre de décision ISW IBM
msassoc	Règles d'association MS
msbayes	MS Naive Bayes
mscluster	Classification non supervisée MS
mslogistic	Régression logistique MS
msneuralnetwork	Réseau neuronal MS
msregression	Régression linéaire MS
mssequencecluster	MS Sequence Clustering
mstimeseries	Séries temporelles MS
mstree	Arbre décision MS
oraabn	Oracle Adaptive Bayes
oraai	Oracle AI
oraapriori	Apriori Oracle
oradecisiontree	Arbre décision Oracle
oraglm	Oracle GLM
orakmeans	Oracle <i>k</i> -Means
oramdl	MDL Oracle
oranb	Oracle Naive Bayes
oranmf	NMF Oracle
oraocluster	O-Cluster Oracle
orasvm	Oracle SVM

Pour éviter les noms de modèle en double

Lorsque vous utilisez des scripts pour manipuler les modèles générés, gardez à l'esprit que les noms de modèle en double peuvent donner lieu à des références ambiguës. Pour éviter ce problème, il s'avère judicieux, lors de la génération de scripts, d'utiliser des noms uniques pour les modèles générés.

Pour définir les options des noms de modèle en double, procédez comme suit :

- ▶ A partir des menus, sélectionnez :
Outils > Options utilisateur
- ▶ Cliquez sur l'onglet Notifications.

- Sélectionnez Remplacer le modèle précédent afin de limiter les noms en double pour les modèles générés.

Le comportement de l'exécution du script peut varier entre SPSS Modeler et IBM SPSS Collaboration and Deployment Services lorsqu'il existe des références de modèle ambiguës. Le client SPSS Modeler inclut l'option « Remplacer le modèle précédent » qui remplace automatiquement les modèles qui ont le même nom (par exemple, lorsqu'un script parcourt une boucle pour produire un modèle différent à chaque fois). Cependant, cette option n'est pas disponible lorsque le même script est exécuté dans IBM SPSS Collaboration and Deployment Services. Vous pouvez éviter cette situation en renommant le modèle généré dans chaque itération pour éviter des références ambiguës aux modèles ou en effaçant le modèle actuel (par exemple en ajoutant une instruction `clear generated palette`) avant la fin de la boucle.

delete model MODEL

```
delete model MODEL
```

Supprime un modèle donné de la palette de nuggets de modèle (ou efface tous les modèles).

```
delete model Drug
```

```
delete model Drug:c50
```

Pour supprimer le dernier modèle inséré par le script en cours, exécutez la commande suivante :

```
delete last model
```

Pour que cette dernière instruction fonctionne, l'instruction `insert model` doit avoir été exécutée au moins une fois au cours de l'exécution du script actuel.

Pour effacer tous les nuggets de modèle dans la palette Modèles, exécutez la commande suivante :

```
clear generated palette
```

export model MODEL as FILE

```
export model MODEL in DIRECTORY format FORMAT
```

```
export model MODEL as FILE format FORMAT
```

Exportation en PMML. Pour exporter le modèle généré au format PMML, exécutez la commande suivante :

```
export model Drug in c:/mymodels format pmml
```

```
export model Drug as c:/mymodel.xml format pmml
```

Pour plus d'informations, reportez-vous à la section [Importation et exportation de modèles au format PMML dans le chapitre 10 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*](#).

Export SQL. Pour exporter un modèle généré au format SQL, exécutez la commande suivante :

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

Remarque : La fonctionnalité d'export SQL n'est disponible que pour certains types de modèle. Pour plus d'informations, reportez-vous à la section [Navigation dans les nuggets de modèle dans le chapitre 3 dans *Noeuds de modélisation de IBM SPSS Modeler 15*](#).

Détails relatifs aux modèles. Pour exporter au format HTML ou texte les détails relatifs à un modèle (tels qu'ils apparaissent dans l'onglet Modèle lorsque vous parcourez le nugget de modèle), exécutez la commande suivante :

```
export model Drug as c:\mymodel.htm format html
```

```
export model Drug as c:\mymodel.txt format text
```

Remarque : Ces formats ne sont pas disponibles pour les modèles ne comportant pas d'onglet Modèle.

Récapitulatif du modèle. Pour exporter au format HTML ou texte le récapitulatif du modèle (tel qu'il apparaît dans l'onglet Récapitulatif lorsque vous parcourez le nugget de modèle), exécutez la commande suivante :

```
export model Drug summary in c:/mymodels format html
```

```
export model Drug summary as c:/mymodel.txt format text
```

```
export model 'assocapriori' as 'C:/temp/assoc_apriori' format html
```

Figure 4-3
Onglet Modèle d'association exporté au format HTML

11 fields				
	Conséquence	Antécédent	% de prise en charge	% de confiance
1	frozenmeal	cannedveg and beer	17,766	87,425
2	cannedveg	frozenmeal and beer	18,085	85,882
3	beer	cannedveg and frozenmeal	18,404	84,393
4	frozenmeal	beer and fish	9,043	58,824
5	frozenmeal	beer	31,17	58,02
6	fruitveg	beer and fish	9,043	57,647
7	frozenmeal	cannedveg and fish	9,468	57,303
8	cannedveg	frozenmeal	32,128	57,285
9	frozenmeal	cannedveg	32,234	57,096
10	cannedveg	beer	31,17	56,997
11	cannedveg	frozenmeal and fish	9,574	56,667
12	cannedveg	beer and fish	0,043	56,471

insert model MODEL

```
insert model MODEL
insert model MODEL at X Y
insert model MODEL between NODE1 and NODE2
insert model MODEL connected between NODE1 and NODE2
```

Ajoute le modèle au flux en cours. Vous pouvez également, si vous le souhaitez, indiquer une position et des options de connexion.

```
insert model Kohonen between :typenode and :analysisnode
```

```
insert model Drug:neuralnetwork connected between 'Define Types' and 'Analysis'
```

load model FILENAME

```
load model FILENAME
```

Charge un modèle enregistré dans la palette Modèles.

```
load model c:/mymodel.gm
```

retrieve model REPOSITORY_PATH

```
retrieve model REPOSITORY_PATH {label LABEL | version VERSION}
```

Récupère un modèle sauvegardé dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
retrieve model "/my folder/Kohonen.gm"
```

save model MODEL as FILENAME

```
save model MODEL as FILENAME
```

Enregistre le modèle indiqué en tant que fichier de modèle généré.

```
save model Drug as c:/mymodel.gm
```

store model MODEL as REPOSITORY_PATH

```
store model MODEL as REPOSITORY_PATH {label LABEL}
```

Stocke le modèle spécifié dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
store model Kohonen as "/my folder/Kohonen.gm"
```

L'extension (*.gm) est facultative, mais doit être utilisée de façon cohérente lors de l'enregistrement et de l'extraction du modèle. Par exemple, si le modèle est stocké simplement sous le nom « Kohenen », il devra être extrait sur la base du même nom. (En d'autres termes, l'extension, si elle est utilisée, fait simplement partie du nom du modèle.)

Objets de flux

Les commandes de script disponibles pour les objets de flux sont les suivantes.

create stream DEFAULT_FILENAME

```
create stream DEFAULT_FILENAME
```

Scripts autonomes uniquement. Crée en mémoire un flux doté du nom indiqué. Le flux n'est pas enregistré automatiquement.

```
create stream 'Druglearn'
```

close STREAM

```
close STREAM
```

Scripts autonomes uniquement. Ferme le flux indiqué.

Pour fermer le flux en cours, entrez la commande suivante tout en minuscules :

```
close stream
```

Scripts autonomes

Si vous utilisez plusieurs flux, notez que le terme `stream` (en minuscules) est une variable spéciale faisant référence au flux en cours. Pour fermer un autre flux, vous pouvez réinitialiser la valeur de cette variable :

```
set stream = get stream 'Stream5'  
close stream
```

Il est également possible de définir toute variable déclarée qui fait référence à un flux ; par exemple :

```
var s  
set s = get stream 'Stream2'  
save s as c:/stream2.str  
close s
```

Enfin, vous pouvez réaffecter le flux en cours de manière temporaire à l'aide de la commande `with stream` :

```
with stream 'Stream1'  
close stream  
endwith
```


clear stream

```
clear stream
```

Supprime tous les noeuds du flux en cours.

get stream STREAM

```
get stream STREAM
```

Scripts autonomes uniquement. Utilisée pour obtenir une référence au flux indiqué, lequel peut être affecté à une variable locale (ou à la variable spéciale stream). Le flux concerné doit déjà être ouvert.

```
var s  
set s = get stream 'Druglearn'  
close s
```

load stream FILENAME

```
load stream FILENAME
```

Scripts autonomes uniquement. Ajoute le flux indiqué à l'espace de travail, et ce sans effacer les noeuds du flux en cours.

```
load stream "c:/demos/druglearn.str"
```

Commande open stream et commande load stream. La commande load stream ajoute le flux indiqué à l'espace de travail, et ce sans effacer les noeuds du flux en cours. Cette commande, très fréquemment utilisée dans les versions précédentes de IBM® SPSS® Modeler, a été largement supplantée dans les nouvelles versions par la possibilité d'ouvrir, de gérer et de copier des noeuds entre plusieurs flux.

open stream FILENAME

```
open stream FILENAME
```

Scripts autonomes uniquement. Ouvre le flux indiqué.

```
open stream "c:/demos/druglearn.str"
```

retrieve stream REPOSITORY_PATH

```
retrieve stream REPOSITORY_PATH {label LABEL | version VERSION}  
retrieve stream URI [{#m.marker | #l.label}]
```

Récupère le flux spécifié dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
retrieve stream "/myfolder/druglearn.str"
```

```
retrieve stream "spsscr:///models/drug%20model.gm#m.0:2005-10-12%2014:15:41.281"
```

save *STREAM* as *FILENAME*

```
save STREAM
save STREAM as FILENAME
```

Pour enregistrer les modifications apportées au flux en cours (en supposant que ce dernier a été enregistré précédemment), entrez la commande suivante tout en minuscules :

```
save stream
```

Pour enregistrer un flux pour la première fois sous un nouveau nom de fichier, entrez la commande suivante :

```
create stream nifty
create featureselectionnode
save stream as c:/nifty.str
```

Scripts autonomes

Si vous utilisez plusieurs flux dans un script autonome, notez que le terme `stream` (en minuscules comme ci-dessus) est une variable spéciale faisant référence au flux en cours. Pour enregistrer un autre flux, vous pouvez réinitialiser la valeur de cette variable :

```
set stream = get stream 'Stream5'
save stream
```

Il est également possible de définir toute variable déclarée qui fait référence à un flux ; par exemple :

```
var s
set s = get stream 'Stream2'
save s as c:/stream2.str
close s
```

Enfin, vous pouvez réaffecter le flux en cours de manière temporaire à l'aide de la commande `with stream` :

```
with stream 'Stream1'
save stream
endwith
```

[Pour plus d'informations, reportez-vous à la section Utilisation de plusieurs flux dans le chapitre 3 sur p. 25.](#)

store stream as *REPOSITORY_PATH*

```
store stream as REPOSITORY_PATH {label LABEL}
store stream as URI [#l.label]
```

```
store stream as "/folder_1/folder_2/mystream.str"
```

Stocke le flux en cours dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
store stream as "/folder_1/folder_2/druglearn.str"
store stream as "spsscr:///folder_1/folder_2/mystream.str"
```

Scripts autonomes

Si vous utilisez plusieurs flux dans un script autonome, notez que le terme `stream` (en minuscules comme ci-dessus) est une variable spéciale faisant référence au flux en cours. Pour stocker un autre flux, vous pouvez réinitialiser la valeur de cette variable :

```
set stream = get stream 'Stream5'
store stream as "/folder_1/mystream.str"
```

De la même manière, toute variable déclarée qui fait référence à un flux peut être indiquée ou le flux actuel peut être temporairement réaffecté à l'aide de la commande `with stream` :

```
with stream 'Stream6'
  store stream as "/folder_1/mystream.str"
endwith
```

with stream STREAM

```
with stream STREAM
  STATEMENTS
endwith
```

Scripts autonomes uniquement. Exécute les `STATEMENTS` avec le `STREAM` indiqué étant défini comme flux en cours. Le flux actuel d'origine est restauré une fois que les instructions ont été exécutées.

```
with stream 'druglearn'
  create typenode
  execute_script
endwith
```

Objets de projet

Les commandes de script disponibles pour les objets de projet sont les suivantes.

L'extension (`*.cpj`) est facultative, mais doit être utilisée de façon cohérente lors du stockage et de l'extraction d'un projet donné.

execute_project

```
execute_project
```

Génère le rapport par défaut du projet en cours.

load project FILENAME

```
load project FILENAME
```

Ouvre le projet spécifié.

```
load project "C:/clemdata/DrugData.cpj"  
set ^project.summary="Initial modeling work on the latest data."  
set ^project.ordering=NameAddedType  
execute_project
```

retrieve project REPOSITORY_PATH

```
retrieve project REPOSITORY_PATH {label LABEL | version VERSION}
```

Récupère un projet dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
retrieve project "/CRISPD/DrugExample.cpj"
```

save project as FILENAME

```
save project  
save project as FILENAME
```

Enregistre le projet en cours.

store project as REPOSITORY_PATH

```
store project as REPOSITORY_PATH {label LABEL}
```

Stocke le projet en cours dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
store project as "/CRISPD/DrugExample.cpj"
```

Objets d'état

Vous pouvez charger un état enregistré à l'aide de la commande `load state`.

load state FILENAME

```
load state FILENAME
```

Charge l'état indiqué.

```
load state "c:/data/myproject.cst"
```

Objets de résultat

Vous pouvez accéder aux résultats à l'aide de la commande `value`.

value RESULT

```
value RESULT at ROW COLUMN
```

Les noeuds terminaux comprennent un paramètre en lecture seule, appelé `output`, qui permet d'accéder au dernier objet généré. Pour les noeuds qui génèrent une sortie tabulaire sous forme de lignes et de colonnes, il est ainsi possible d'accéder à la valeur d'une cellule donnée ; par exemple :

```
execute :tablenode
set last_row = :tablenode.output.row_count
set last_column = :tablenode.output.column_count
set last_value = value :tablenode.output at ^last_row ^last_column
var myresults
set myresults = open create 'C:/myresults.txt'
write myresults 'The value in the last cell is ' >< ^last_value
```

La numérotation des lignes et des colonnes commence à partir de la valeur 1. Si l'objet de sortie n'existe pas, une erreur est renvoyée.

Propriétés des objets de résultat

Les propriétés suivantes sont communes aux objets de résultat (tels que les résultats de Table et de Matrice) qui intègrent les données dans des lignes et des colonnes :

Propriété	Description
<code>row_count</code>	Renvoie le nombre de lignes des données.
<code>column_count</code>	Renvoie le nombre de colonnes des données.

Objets de fichier

Les commandes de script disponibles pour les objets de fichier sont les suivantes.

close FILE

```
close FILE
```

L'instruction ci-dessus ferme le fichier indiqué.

open FILE

```
open create FILENAME
open append FILENAME
```

Les instructions ci-dessus ouvrent le fichier indiqué.

- **créer.** Crée le fichier s'il n'existe pas ou l'écrase s'il existe.
- **ajouter.** S'ajoute à un fichier existant. Génère une erreur si le fichier n'existe pas.

Cette opération renvoie le descripteur de fichier correspondant au fichier ouvert.

```
var file
set file = open create 'C:/script.out'
for I from 1 to 3
  write file 'Stream ' >< I
endfor
close file
```

write FILE

```
write FILE TEXT_EXPRESSION
writeln FILE TEXT_EXPRESSION
```

Ces expressions inscrivent le texte TEXT_EXPRESSION dans le fichier. La première instruction inscrit le texte tel quel ; la seconde crée également une ligne après l'écriture de l'expression. Une erreur est générée si FILE n'est pas un fichier ouvert.

```
var file
set file = open create 'C:/hello.txt'
writeln file 'Hello'
writeln file 'World'
write file 'Would you like to play a game?'
close file
```

Objets de sortie

Les commandes de script disponibles pour les objets de sortie sont les suivantes.

Nom des types de sortie

Le tableau suivant répertorie tous les types d'objet de sortie et les noeuds qui les génèrent. Pour obtenir la liste complète des formats d'exportation disponibles pour chaque type d'objet de sortie, reportez-vous à la description des propriétés propres au noeud qui génère le type de sortie en question, dans [le chapitre 15, Propriétés des noeuds Graphiques](#), et [le chapitre 19, Propriétés des noeuds de sortie](#).

Type d'objet de sortie	Noeud
analysisoutput	Analyse
collectionoutput	Résumé
dataauditoutput	Audit données
distributionoutput	Proportion
evaluationoutput	Evaluation
histogramoutput	Histogramme

Type d'objet de sortie	Noeud
matrixoutput	Matrice
meansoutput	Moyennes
multiplotoutput	Courbes
plotoutput	Nuage
qualityoutput	Qualité
reportdocumentoutput	Ce type d'objet n'est pas issu d'un noeud ; il s'agit en fait de la sortie créée par un rapport de projet.
reportoutput	Rapport
statisticsprocedureoutput	StatisticsSortie
statisticsoutput	Statistiques
tableoutput	Table
timeplotoutput	Tracé horaire
weboutput	Relations

delete output OUTPUT

delete output OUTPUT

Supprime la sortie indiquée de la palette du gestionnaire. Par exemple :

```
delete output :statisticsoutput
```

Pour supprimer tous les éléments de sortie de la palette du gestionnaire, exécutez la commande suivante :

```
clear outputs
```

export output OUTPUT

export output OUTPUT as FILE format FORMAT

Exporte la sortie au format indiqué. Les formats disponibles dépendent du type de sortie ; ils doivent être identiques à ceux accessibles dans le menu Exporter lorsque vous parcourez la sortie indiquée.

```
export output :statisticsoutput as "C:/output/statistics.html" format html
export output :matrixoutput as "C:/output/matrix.csv" format delimited
export output :tableoutput as "C:/output/table.tab" format transposed formatted
```

get output OUTPUT

get output OUTPUT

Obtient une référence à la sortie indiquée. Vous pouvez, par exemple, utiliser une boucle pour obtenir une série d'objets de sortie et exporter les objets un par un.

```
execute_all
for item in statisticsoutput matrixoutput tableoutput
```

```
var theoutput
set theoutput = get output :^item
set filename = 'c:/><^item ><'.htm'
export output ^theoutput as ^filename format html
endfor
```

load output FILENAME

```
load output FILENAME
```

Charge la sortie indiquée.

```
load output 'c:/matrix.cou'
```

retrieve output REPOSITORY_PATH

```
retrieve output REPOSITORY_PATH {label LABEL | version VERSION}
```

Récupère la sortie spécifiée dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
retrieve output "/results/mytable"
```

save output OUTPUT as FILENAME

```
save output as FILENAME
```

Enregistre la sortie indiquée.

```
save output :matrixoutput as 'c:/matrix.cou'
```

store output OUTPUT as REPOSITORY_PATH

```
store output OUTPUT as REPOSITORY_PATH {label LABEL}
```

Stocke la sortie spécifiée dans le IBM® SPSS® Collaboration and Deployment Services Repository. [Pour plus d'informations, reportez-vous à la section Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60.](#)

```
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

```
store output :tableoutput as "/results/mytable"
```


Conseils pour la génération de scripts

Cette section présente les différents conseils et techniques pour l'utilisation des scripts : modification de l'exécution du flux, utilisation d'un mot de passe codé dans un script et accès aux objets du IBM® SPSS® Collaboration and Deployment Services Repository.

Modification de l'exécution du flux

Lorsqu'un flux est exécuté, ses noeuds terminaux sont exécutés dans un ordre optimisé pour la situation par défaut. Dans certains cas, vous pouvez choisir un ordre d'exécution différent. Pour modifier l'ordre d'exécution d'un flux, réalisez les opérations suivantes dans l'onglet Script de la boîte de dialogue Propriétés du flux :

- ▶ Commencez par un script vide.
- ▶ Cliquez sur le bouton Ajouter le script par défaut de la barre d'outils pour ajouter le script de flux par défaut.
- ▶ Modifiez à votre convenance l'ordre des instructions qu'il contient.

Bouclage dans les noeuds

Vous pouvez utiliser une boucle for en association avec la propriété `^stream.nodes` pour boucler tous les noeuds d'un flux. Par exemple, le script suivant boucle tous les noeuds et met les noms de champ de tous les noeuds Filtrer en majuscules.

Ce script peut être utilisé dans n'importe quel flux comportant un noeud, même si aucun champ n'est en réalité filtré. Ajoutez simplement un noeud Filtrer qui transmet tous les champs afin de mettre les noms de champ en majuscules sur l'ensemble du système.

```
var my_node
var loop_me
var var_name

for my_node in ^stream.nodes
  if ^my_node.node_type = filternode then
    for loop_me in _fields_to ^my_node:filternode
      set var_name = lowertoupper(^my_node:filternode.new_name.^loop_me)
      set ^my_node:filternode.new_name.^loop_me = ^var_name
    endfor
  else
  endif
endfor
```

Le script boucle tous les noeuds du flux actuel, tels qu'ils sont renvoyés par la propriété `^stream.nodes` et vérifie si chaque noeud est un noeud Filtrer. Si tel est le cas, le script boucle chaque champ dans le noeud et utilise la fonction `lowertoupper()` pour mettre le nom en majuscules.

Astuce : Pour mettre les noms de champ en minuscules, utilisez la fonction `uppertolower()`.

Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository

Remarque : Une licence distincte est requise pour accéder à un référentiel IBM® SPSS® Collaboration and Deployment Services. Pour plus d'informations, reportez-vous à <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Si vous disposez d'une licence pour le référentiel IBM® SPSS® Collaboration and Deployment Services Repository, vous pouvez stocker, verrouiller et déverrouiller des objets dans le référentiel et en extraire à l'aide des commandes de script. Le référentiel permet de gérer le cycle de vie des modèles de Data mining et des objets prédictifs associés dans le contexte d'applications, de solutions et d'outils professionnels. [Pour plus d'informations, reportez-vous à la section A propos du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 9 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Connexion au IBM SPSS Collaboration and Deployment Services Repository

Pour pouvoir accéder au référentiel, vous devez dans un premier temps paramétrer une connexion valide avec ce dernier, en utilisant le menu Outils de l'interface utilisateur IBM® SPSS® Modeler ou avec une ligne de commande. ([Pour plus d'informations, reportez-vous à la section IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion dans le chapitre 7 sur p. 74.](#))

Stockage et extraction des objets

Au sein d'un script, les commandes `retrieve` et `store` vous permettent d'accéder à différents objets, notamment les flux, modèles, sorties, noeuds et projets. La syntaxe est la suivante :

```
store object as REPOSITORY_PATH {label LABEL}
store object as URI [#l.label]
```

```
retrieve object REPOSITORY_PATH {label LABEL | version VERSION}
retrieve object URI [#m.marker | #l.label]
```

Le `REPOSITORY_PATH` fournit l'emplacement de l'objet dans le référentiel. Le chemin doit être placé entre guillemets et délimité par des barres obliques. Il ne distingue pas les majuscules des minuscules.

```
store stream as "/folder_1/folder_2/mystream.str"
store model Drug as "/myfolder/drugmodel"
store model Drug as "/myfolder/drugmodel.gm" label "final"
store node DRUG1n as "/samples/drug1ntypenode"
store project as "/CRISPDM/DrugExample.cpj"
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

Il est également possible d'inclure dans le nom de l'objet une extension telle que *.str* ou *.gm*, mais cela n'est pas obligatoire tant que le nom est cohérent. Par exemple, si un modèle est stocké sans extension, il doit être extrait sur la base du même nom :

```
store model "/myfolder/drugmodel"
retrieve model "/myfolder/drugmodel"
```

et

```
store model "/myfolder/drugmodel.gm"
retrieve model "/myfolder/drugmodel.gm" version "0:2005-10-12 14:15:41.281"
```

Lors de l'extraction d'objets, la version la plus récente de l'objet est toujours renvoyée, sauf si vous indiquez une version ou une étiquette particulière. Lors de l'extraction d'un objet de noeud, le noeud est automatiquement inséré dans le flux actuel. Lors de l'extraction d'un objet de flux, vous devez utiliser un script autonome. Vous ne pouvez pas extraire un objet de flux à partir d'un script de flux.

Verrouillage et déverrouillage d'objets

A partir d'un script, vous pouvez verrouiller un objet pour empêcher les autres utilisateurs de mettre à jour ses versions existantes ou de créer de nouvelles versions. Vous pouvez également déverrouiller un objet que vous avez verrouillé.

La syntaxe pour verrouiller et déverrouiller un objet est la suivante :

```
lock REPOSITORY_PATH
lock URI
```

```
unlock REPOSITORY_PATH
unlock URI
```

Tout comme avec le stockage et la récupération d'objets, le `REPOSITORY_PATH` vous donne l'emplacement de l'objet dans le référentiel. Le chemin doit être placé entre guillemets et délimité par des barres obliques. Il ne distingue pas les majuscules des minuscules.

```
lock "/myfolder/Stream1.str"
```

```
unlock "/myfolder/Stream1.str"
```

Vous pouvez également utiliser un URI (Uniform Resource Identifier) plutôt qu'un chemin de référentiel pour fournir l'emplacement de l'objet. L'URI doit inclure le préfixe `spsscr:` et doit être entièrement entre guillemets. Seules les barres obliques sont autorisées en tant que séparateurs de chemin ; les espaces, quant à eux, doivent être codés. En d'autres termes, utilisez `%20` à la place d'un espace dans le chemin. L'URI ne distingue pas les majuscules des minuscules. Voici quelques exemples :

```
lock "spsscr:///myfolder/Stream1.str"
```

```
unlock "spsscr:///myfolder/Stream1.str"
```

Veillez noter que le verrouillage d'objets s'applique à toutes les versions d'un objet. Vous ne pouvez pas verrouiller ou déverrouiller des versions individuelles.

Génération d'un mot de passe codé

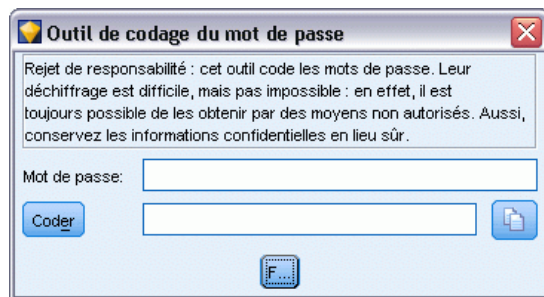
Dans certains cas, vous pouvez être amené à ajouter un mot de passe à un script, par exemple, pour accéder à une source de données protégée par un mot de passe. Vous pouvez utiliser les mots de passe codés dans :

- Les propriétés des noeuds source et de sortie SGBD
- Les arguments de ligne de commande permettant la connexion au serveur
- Les propriétés de connexion à la base de données stockées dans un fichier *.par* (fichier de paramètres généré depuis l'onglet Publier d'un noeud Export)

L'interface utilisateur dispose d'un outil qui permet de générer des mots de passe codés à partir de l'algorithme Blowfish (pour plus de détails, consultez le site <http://www.schneier.com/blowfish.html>). Une fois le mot de passe codé, vous pouvez le copier et le stocker dans des fichiers de script et des arguments de ligne de commande. Le `epassword` de la propriété du noeud utilisé pour `databasenode` et `databaseexportnode` stocke le mot de passe codé.

- Pour générer un mot de passe codé, choisissez l'option suivante dans le menu Outils :
Coder le mot de passe...

Figure 5-1
Outil de codage du mot de passe



- Entrez un mot de passe dans la zone de texte Mot de passe.
- Cliquez sur Coder pour générer le codage aléatoire de votre mot de passe.
- Cliquez sur le bouton Copier pour copier le mot de passe codé dans le Presse-papiers.
- Collez le mot de passe dans le script ou le paramètre souhaité.

Vérification du script

Vous pouvez procéder à une vérification rapide de la syntaxe de tous les types de script en cliquant sur le bouton de vérification rouge dans la barre d'outils de la boîte de dialogue Script autonome.

Figure 5-2
Icônes de la barre d'outils du script de flux



La vérification du script vous prévient de toute erreur se produisant dans votre code et vous propose des recommandations d'amélioration. Pour visualiser la ligne comportant des erreurs, cliquez sur le commentaire dans la partie inférieure de la boîte de dialogue. L'erreur est surlignée en rouge.

Génération de scripts à partir de la ligne de commande

La génération de scripts vous permet d'exécuter des opérations généralement effectuées dans l'interface utilisateur. Il vous suffit d'indiquer et d'exécuter un flux autonome dans la ligne de commande lors du lancement de IBM® SPSS® Modeler. Par exemple :

```
client -script scores.txt -execute
```

Le commutateur `-script` charge le script spécifié et le commutateur `-execute` exécute toutes les commandes du fichier de script.

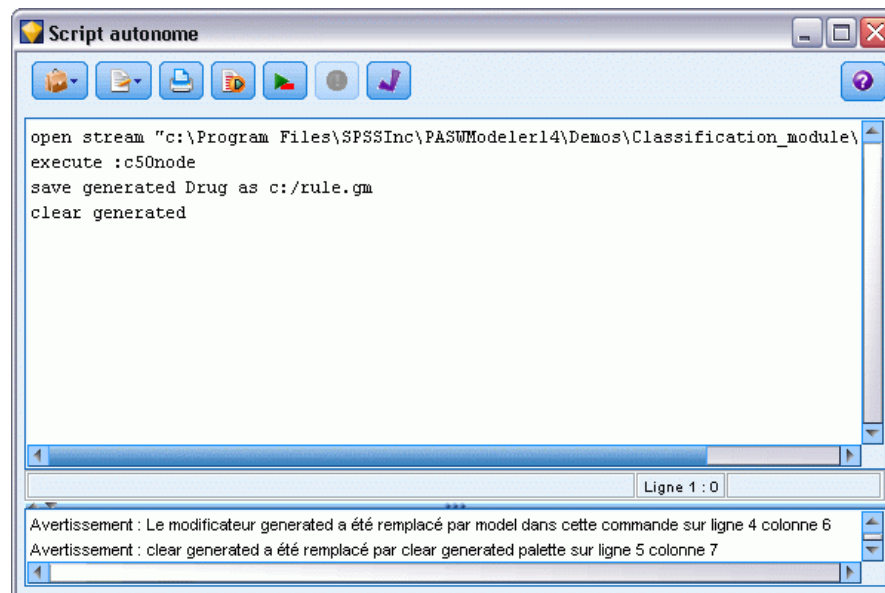
Compatibilité avec les versions précédentes

Les scripts créés dans les versions antérieures de IBM® SPSS® Modeler doivent en général fonctionner tels quels dans la nouvelle version. Cependant, les nuggets de modèle peuvent désormais être insérés automatiquement dans le flux (réglage par défaut), et peuvent soit remplacer soit compléter un nugget existant de ce type dans le flux. Ceci dépend des réglages des options Ajout de modèle à un flux et Remplacer le modèle précédent (Outils > Options > Options utilisateur > Notifications). Vous pouvez, par exemple, être amené à modifier le script d'une version précédente dans laquelle le remplacement de nugget est géré en supprimant le nugget existant et en insérant le nouveau.

Il se peut que des scripts créés dans la version actuelle ne fonctionnent pas dans les versions antérieures.

Si un script créé dans une version antérieure utilise une commande qui a été remplacée depuis (ou qui est obsolète), l'ancienne forme reste prise en charge mais un message d'avertissement apparaît. Par exemple, l'ancien mot de passe `generated` a été remplacé par `model`, et `clear generated` a été remplacé par `clear generated palette`. Les scripts qui utilisent les anciennes formes continuent de s'exécuter mais un avertissement apparaît.

Figure 5-3
Exécution d'un script qui utilise une commande obsolète



Exemples de génération de scripts

Cette section présente des exemples indiquant comment les scripts peuvent être utilisés dans IBM® SPSS® Modeler.

Rapport de noeud Typer

Ce script crée un rapport HTML répertoriant des informations sur les champs du flux actuel. Le script peut être utilisé avec tous les flux ayant un noeud Typer instancié et il peut être facilement étendu pour couvrir des propriétés ou des noeuds supplémentaires.

- Les balises HTML standard sont utilisées pour formater les résultats et les afficher dans un navigateur classique.
- Un noeud Typer IBM® SPSS® Modeler est utilisé pour accéder aux propriétés de chaque champ. Le script peut facilement être étendu pour répertorier des propriétés supplémentaires présentées via le noeud Typer, par exemple les valeurs manquantes ou le rôle du champ. [Pour plus d'informations, reportez-vous à la section Propriétés de typenode dans le chapitre 14 sur p. 176.](#)
- Les commandes de script SPSS Modeler sont utilisées pour écrire la sortie dans un fichier et pour parcourir des champs en boucle afin d'accéder aux propriétés de chacun. [Pour plus d'informations, reportez-vous à la section Commandes de script dans le chapitre 4 sur p. 32.](#)

Figure 6-1
Exemple de script de rapport de noeud Typer

```
# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"
writeln myreport "<header>Informations de IBM SPSS Modeler sur le noeud Typer</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border='1'"
writeln myreport "<tr bgcolor='C0C0C0'"
writeln myreport "<td>Champ</td><td>Type</td><td>Valeurs</td>"
writeln myreport "</tr>"

# loop through fields and add a row for each
var current_field
for current_field in _fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >> ^current_field >> "</td>"
  write myreport "<td>" >> Type:typenode.type.^current_field >> "</td>"
```

```

# add values for numeric fields
if Type:typenode.type.^current_field = Range then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif

# add values for flag fields
if Type:typenode.type.^current_field = Flag then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif

# add values for nominal fields
if Type:typenode.type.^current_field = Set then
  writeln myreport "<td>"
  var current_value
  for current_value in Type:typenode.values.^current_field
    writeln myreport ^current_value >< "<BR/>"
  endfor
  writeln myreport "</td>"
endif

writeln myreport "</tr>"
endfor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport

```

Création du fichier de sortie

Le script commence par créer un fichier HTML et ajoute les balises nécessaires pour générer un tableau avec une ligne d'en-tête répertoriant les titres de colonne *Champ*, *Type* et *Valeurs*. (Chaque paire de balises <td> </td> crée une cellule dans une ligne de tableau.) Ces colonnes seront renseignées pour chaque champ en fonction des propriétés du noeud *Type*.

```

# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"
writeln myreport "<header>Informations de IBM SPSS Modeler sur le noeud Type</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border='1'">"
writeln myreport "<tr bgcolor='COCOC0'">"
writeln myreport "<td>Champ</td><td>Type</td><td>Valeurs</td>"
writeln myreport "</tr>"

```


Bouclage dans les champs

Ensuite, le script parcourt tous les champs en boucle dans le noeud `Type` et ajoute une ligne pour chaque champ répertoriant le nom et le type de champ.

```
# loop through fields and add a row for each
var current_field
for current_field in _fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >< ^current_field >< "</td>"
  write myreport "<td>" >< Type:typenode.type.^current_field >< "</td>"
```

Valeurs pour les champs continus et les champs booléens

Pour les champs continus (intervalle numérique), la propriété `typenode.values` renvoie les valeurs inférieure et supérieure au format `[0.500517, 0.899774]` affiché dans le tableau. Pour les champs booléens, les valeurs `True/False` sont affichées dans un format similaire.

```
# add values for numeric fields
if Type:typenode.type.^current_field = Range then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif

# add values for flag fields
if Type:typenode.type.^current_field = Flag then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif
```

Valeurs des champs nominaux.

Pour les champs nominaux, la propriété `typenode.values` renvoie la liste complète des valeurs définies. Le script parcourt cette liste en boucle pour chaque champ afin d'insérer chaque valeur tour à tour, avec un saut de ligne (balise `
`) entre chacune d'elles.

```
# add values for nominal fields
if Type:typenode.type.^current_field = Set then
  writeln myreport "<td>"
  var current_value
  for current_value in Type:typenode.values.^current_field
    writeln myreport ^current_value >< "<BR/>"
  endfor
  writeln myreport "</td>"
endif
```

Fermeture du fichier

En dernier lieu, le script ferme la ligne, les balises `<table>`, `<body>` et `<html>`, et le fichier de sortie.

```
writeln myreport "</tr>"
endfor
```

```
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport
```

Rapport de flux

Ce script crée un rapport HTML répertoriant, pour chaque noeud, le nom, le type et l'annotation dans le flux actuel. Outre les règles de base de la création d'un fichier HTML et de l'accès aux propriétés de noeud et de flux, il démontre comment créer une boucle qui exécute un ensemble particulier d'instructions pour chaque noeud dans un flux. Il peut être utilisé avec tous les flux.

Figure 6-2

Exemple de script de rapport de flux

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile "<BODY>"
writeln myfile "<HEAD>Report for stream " >> ^stream.name >> ".str</HEAD>"
writeln myfile "<p>" >> ^stream.annotation >> "</p>"

#Create the table with header row
writeln myfile "<TABLE border=\\"1\" width=\\"90%\">"
writeln myfile " <tr bgcolor=\\"lightgrey\" colspan=\\"3\">"
writeln myfile " <th>Node Name</th>"
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile " </tr>"

# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.name
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.node_type
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.annotation >> "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
endfor

writeln myfile "</TABLE>"
writeln myfile "</BODY>"
```

```
writeln myfile "</HTML>"
close myfile
```

Création du rapport

Le script commence par créer un fichier HTML avec des éléments <BODY> et <HEAD>. La propriété `^stream.name` renvoie le nom du flux actuel, qui est inséré dans l'en-tête. L'opérateur `><` est utilisé pour concaténer plusieurs chaînes.

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile "<BODY>"
writeln myfile "<HEAD>Report for stream " >> ^stream.name >> ".str</HEAD>"
writeln myfile "<p>" >> ^stream.annotation >> "</p>"
```

Ensuite, le script crée un tableau HTML avec une ligne d'en-tête répertoriant les titres de colonne *Nom du noeud*, *Type* et *Annotation*. (Chaque paire de balises `<td></td>` crée une cellule dans une ligne de tableau.)

```
#Create the table with header row
writeln myfile "<TABLE border=\\"1\" width=\\"90%\">"
writeln myfile "<tr bgcolor=\\"lightgrey\" colspan=\\"3\">"
writeln myfile " <th>Node Name</th>"
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile "</tr>"
```

Ensuite, le script parcourt en boucle tous les noeuds du flux actuel. Une ligne est ajoutée au tableau pour chaque noeud, répertoriant le nom, le type et l'annotation. Un espace insécable (` `) est inséré après l'annotation pour éviter de créer une cellule vide si aucune annotation n'est indiquée pour un noeud donné. (Les cellules vides peuvent entraîner un formatage imprévu lors de l'affichage de la table.)

```
# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.name"
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.node_type"
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.annotation >> "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
endfor
```

Enfin, le script ajoute les balises HTML nécessaires à la fermeture du document et ferme le fichier.

```
writeln myfile "</TABLE>"  
writeln myfile "</BODY>"  
writeln myfile "</HTML>"  
close myfile
```

Arguments de ligne de commande

Appel du logiciel

Vous pouvez utiliser la ligne de commande de votre système d'exploitation pour lancer IBM® SPSS® Modeler comme suit :

- ▶ Dans le cas d'un ordinateur sur lequel est installé IBM® SPSS® Modeler, ouvrez une fenêtre DOS ou une invite de commande.
- ▶ Pour lancer l'interface SPSS Modeler en mode interactif, tapez la commande `modelerclient` suivie des arguments souhaités, par exemple :

```
modelerclient -stream report.str -execute
```

Les arguments disponibles (drapeaux) vous permettent de vous connecter à un serveur, de charger des flux, d'exécuter des scripts, ou d'indiquer les autres paramètres nécessaires.

Utilisation d'arguments de ligne de commande

Vous pouvez ajouter des arguments de ligne de commande (également connus sous le nom de **commutateurs**) à la commande `modelerclient` initiale pour modifier l'appel de IBM® SPSS® Modeler.

Vous pouvez par exemple utiliser les commutateurs `-server`, `-stream` et `-execute` pour vous connecter à un serveur, puis charger et exécuter un flux, comme indiqué ci-dessous :

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

Notez que lors d'une exécution en parallèle avec l'installation d'un client local, les arguments de connexion au serveur ne sont pas obligatoires.

Il est possible de placer entre guillemets doubles — les valeurs de paramètre qui contiennent des espaces. Par exemple :

```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

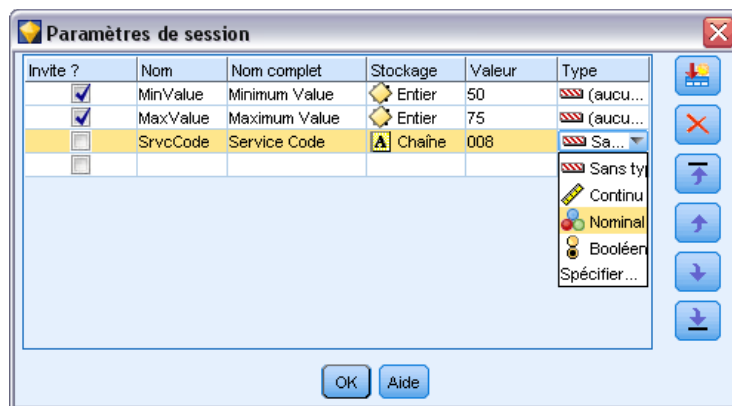
Vous pouvez également exécuter des états et des scripts SPSS Modeler de la même manière, en utilisant respectivement les commutateurs `-state` et `-script`.

Débogage d'arguments de ligne de commande

Pour déboguer une ligne de commande, utilisez la commande `modelerclient` afin de lancer SPSS Modeler avec les arguments souhaités. Vous avez ainsi la possibilité de vérifier que les commandes s'exécutent comme souhaité. Vous pouvez également vérifier les valeurs de

paramètre transmises depuis la ligne de commande dans la boîte de dialogue Paramètres de session (menu Outils, Définir les paramètres de session).

Figure 7-1
Définition des paramètres pour la session



Combinaison de plusieurs arguments

Utilisez le symbole @ suivi du nom du fichier pour combiner plusieurs arguments dans un fichier de commande unique défini lors de l'appel de la commande. Vous pouvez ainsi raccourcir l'appel via la ligne de commande et remédier aux éventuelles limites appliquées à la longueur des commandes par les systèmes d'exploitation. Par exemple, la commande de démarrage suivante utilise les arguments spécifiés dans le fichier référencé par <commandFileName>.

```
modelerclient @<commandFileName>
```

Si vous devez utiliser des espaces, placez le nom et le chemin d'accès du fichier entre guillemets, comme indiqué ci-dessous :

```
modelerclient @ "C:\Program Files\IBM\SPSS\Modeler\scripts\my_command_file.txt"
```

Le fichier de commande peut contenir tous les arguments qui étaient auparavant spécifiés séparément au démarrage, avec un argument par ligne. Par exemple :

```
-stream report.str
-Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt
-execute
```

Lorsque vous rédigez des fichiers de commande et que vous y faites référence, veillez à respecter les contraintes suivantes :

- N'inscrivez qu'une commande par ligne.
- N'intégrez pas un argument @CommandFile au sein d'un fichier de commande.

Arguments de connexion au serveur

Le commutateur `-server` indique à IBM® SPSS® Modeler qu'il doit se connecter à un serveur public et les commutateurs `-hostname`, `-use_ssl`, `-port`, `-username`, `-password`, et `-domain` indiquent à SPSS Modeler comment se connecter au serveur public. Si aucun argument `-server` n'est spécifié, le serveur par défaut ou le serveur local est utilisé.

Exemples

Pour vous connecter à un serveur public :

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

Pour vous connecter à un groupe de serveurs :

```
modelerclient -server -cluster "QA Machines" \
-spsscr_hostname pes_host -spsscr_port 8080 \
-spsscr_username asmith -spsscr_epassword xyz
```

Veillez noter que la connexion à un groupe de serveur nécessite le coordinateur de processus avec IBM® SPSS® Collaboration and Deployment Services. Par conséquent, l'argument `-cluster` doit être utilisé conjointement avec les options de connexion du référentiel (`spsscr_*`). [Pour plus d'informations, reportez-vous à la section IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion sur p. 74.](#)

Argument	Comportement/description
<code>-server</code>	Exécute SPSS Modeler en mode serveur, se connecte à un serveur public à l'aide des commutateurs <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> , et <code>-domain</code> .
<code>-hostname <nom></code>	Nom d'hôte du serveur. Disponible en mode serveur uniquement.
<code>-use_ssl</code>	Indique que la connexion doit utiliser le protocole SSL (Secure Socket Layer). Ce commutateur est facultatif ; par défaut, le protocole SSL n'est pas utilisé.
<code>-port <nombre></code>	Numéro de port du serveur spécifié. Disponible en mode serveur uniquement.
<code>-cluster <nom></code>	Spécifie une connexion à un groupe de serveurs plutôt qu'à un serveur nommé ; cet argument est une alternative aux arguments <code>hostname</code> , <code>port</code> et <code>use_ssl</code> . Le nom est le nom de groupe ou une URI unique qui identifie le groupe dans le IBM® SPSS® Collaboration and Deployment Services Repository. Le groupe de serveurs est géré par le coordinateur de processus dans IBM SPSS Collaboration and Deployment Services. Pour plus d'informations, reportez-vous à la section IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion sur p. 74.
<code>-username <nom></code>	Nom d'utilisateur utilisé pour la connexion au serveur. Disponible en mode serveur uniquement.
<code>-password <mot de passe></code>	Mot de passe utilisé pour la connexion au serveur. Disponible en mode serveur uniquement. <i>Remarque</i> : si le commutateur <code>-password</code> n'est pas utilisé, le système vous invite à entrer un mot de passe.
<code>-epassword <encodedpasswordstring></code>	Mot de passe codé utilisé pour la connexion au serveur. Disponible en mode serveur uniquement. <i>Remarque</i> : vous pouvez créer un mot de passe codé à partir du menu Outils de l'application SPSS Modeler.

Argument	Comportement/description
-domain <nom>	Domaine utilisé pour la connexion au serveur. Disponible en mode serveur uniquement.
-P <nom>=<valeur>	Sert à définir un paramètre de démarrage. Peut également être utilisé pour définir les propriétés de noeud (paramètres de propriété).

IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion

Remarque : Une licence distincte est requise pour accéder à un référentiel IBM® SPSS® Collaboration and Deployment Services. Pour plus d'informations, reportez-vous à <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Pour pouvoir stocker ou extraire des objets du IBM SPSS Collaboration and Deployment Services via la ligne de commande, vous devez indiquer une connexion valide au IBM® SPSS® Collaboration and Deployment Services Repository. Par exemple :

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

Le tableau suivant répertorie les arguments qu'il convient d'utiliser pour paramétrer la connexion :

Argument	Comportement/description
-spsscr_hostname <nom d'hôte ou adresse IP>	Nom d'hôte ou adresse IP du serveur sur lequel est installé le IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <nombre>	Numéro de port sur lequel le IBM SPSS Collaboration and Deployment Services Repository accepte les connexions (8080 par défaut).
-spsscr_use_ssl	Indique que la connexion doit utiliser le protocole SSL (Secure Socket Layer). Ce commutateur est facultatif ; par défaut, le protocole SSL n'est <i>pas</i> utilisé.
-spsscr_username <nom>	Nom d'utilisateur utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <mot de passe>	Mot de passe utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <mot de passe codé>	Mot de passe codé utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_domain <nom>	Domaine utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository. Ce commutateur est facultatif—ne l'utilisez que si vous vous connectez via LDAP ou Active Directory.

Arguments système

Le tableau suivant décrit les arguments système disponibles pour l'appel de la ligne de commande de l'interface utilisateur :

Argument	Comportement/description
@ <commandFile>	Le caractère @ suivi d'un nom de fichier indique une liste de commandes. Lorsque modelerclient détecte un argument commençant par le caractère @, il agit sur les commandes du fichier correspondant comme si elles s'étaient trouvées dans la ligne de commande. Pour plus d'informations, reportez-vous à la section Combinaison de plusieurs arguments sur p. 72 .
-directory <dir>	Définit le répertoire de travail par défaut. En mode local, ce répertoire est utilisé pour les données et la sortie.
-server_directory <dir>	Définit le répertoire du serveur par défaut pour les données. Le répertoire de travail, défini à l'aide du commutateur -directory est utilisé pour la sortie.
-execute	Après le démarrage, exécute tout flux, état ou script chargé au démarrage. Si un script est chargé en plus d'un flux ou d'un état, seul le script est exécuté.
-stream <flux>	Au démarrage, charge le flux spécifié. Vous pouvez spécifier plusieurs flux, mais le dernier flux sera défini comme flux en cours.
-script <script>	Charge, au démarrage, le script autonome spécifié. Vous pouvez le spécifier en plus d'un flux ou d'un état comme décrit ci-dessous, mais un seul script peut être chargé au démarrage.
-model <Modèle>	Au démarrage, charge le modèle généré (fichier au format .gm) indiqué.
-state <état>	Au démarrage, charge l'état enregistré indiqué.
-project <projet>	Charge le projet spécifié. Vous ne pouvez charger qu'un seul projet au démarrage.
-output <sortie>	Au démarrage, charge l'objet de sortie enregistré (fichier au format .cou).
-help	Affiche la liste des arguments de ligne de commande. Lorsque cette option est précisée, tous les autres arguments sont ignorés et l'écran Aide apparaît.
-P <nom>=<valeur>	Sert à définir un paramètre de démarrage. Peut également être utilisé pour définir les propriétés de noeud (paramètres de propriété).

Remarque : vous pouvez également définir les répertoires par défaut dans l'interface utilisateur. Pour accéder aux options, dans le menu Fichier, sélectionnez Définir le répertoire de travail ou Définir le répertoire du serveur.

Chargement de plusieurs fichiers

A partir de la ligne de commande, vous pouvez charger plusieurs flux, états et sorties au démarrage en répétant l'argument applicable pour chaque objet chargé. Par exemple, pour charger et exécuter deux flux nommés *report.str* et *train.str*, utilisez la commande suivante :

```
modelerclient -stream report.str -stream train.str -execute
```

Chargement d'objets du IBM SPSS Collaboration and Deployment Services Repository

Etant donné que vous pouvez charger certains objets à partir d'un fichier ou du IBM® SPSS® Collaboration and Deployment Services Repository (si vous disposez de la licence correspondante), le préfixe de nom de fichier spssc: et éventuellement file: (pour les objets sur

disque) indique à IBM® SPSS® Modeler l'emplacement de l'objet. Le préfixe fonctionne avec les commutateurs suivants :

- -stream
- -script
- -output
- -model
- -project

Vous utilisez le préfixe pour créer un URI qui indique l'emplacement de l'objet. Par exemple : -stream "spsscr:///folder_1/scoring_stream.str". Lorsque le préfixe spsscr: est présent, vous devez définir dans la même commande une connexion valide au IBM SPSS Collaboration and Deployment Services Repository. Ainsi, par exemple, la commande complète serait semblable à ce qui suit :

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

Pour plus de détails sur les URI relatifs aux objets du IBM SPSS Collaboration and Deployment Services Repository, reportez-vous à [Accès aux objets du IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 5 sur p. 60](#). A partir de la ligne de commande, vous devez utiliser un URI. L'élément simple REPOSITORY_PATH n'est pas pris en charge. (Il fonctionne uniquement au sein des scripts.)

Arguments de paramètre

Les paramètres peuvent être utilisés en tant que commutateurs pendant l'exécution de la ligne de commande de IBM® SPSS® Modeler. Dans les arguments de ligne de commande, le commutateur -P sert à indiquer un paramètre du type -P <nom>=<valeur>.

Il peut s'agir des paramètres suivants :

- Des **paramètres simples** (ou des paramètres utilisés directement dans les expressions CLEM).
- Des **paramètres de propriété**, également appelés **propriétés de noeud**. Ces paramètres servent à modifier les paramètres des noeuds du flux. [Pour plus d'informations, reportez-vous à la section Présentation des propriétés de noeud dans le chapitre 9 sur p. 118.](#)
- Des **paramètres de ligne de commande**, utilisés pour modifier l'appel de SPSS Modeler.

Par exemple, vous pouvez fournir les noms d'utilisateur et les mots de passe d'une source de données sous la forme d'un commutateur de ligne de commande, comme l'illustre l'exemple suivant :

```
modelerclient -stream response.str -P:databasenode.datasouce={"ORA 10gR2", user1, mypsw, true}
```

Le format est identique à celui du paramètre `datasouce` de la propriété du noeud `databasenode`. [Pour plus d'informations, reportez-vous à la section Propriétés de databasenode dans le chapitre 12 sur p. 128.](#)

Référence du langage CLEM

Présentation de la référence CLEM

Cette section décrit le Langage de contrôle (CLEM) pour la Manipulation d'Expressions, puissant outil permettant d'analyser et de manipuler les données utilisées dans les flux IBM® SPSS® Modeler. Vous pouvez utiliser le langage CLEM au sein des noeuds pour exécuter entre autres les tâches suivantes : évaluation des conditions, calcul de valeurs, insertion de données dans les rapports. [Pour plus d'informations, reportez-vous à la section A propos de CLEM dans le chapitre 7 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Un sous-ensemble du langage CLEM peut également être utilisé lorsque vous générez des scripts dans l'interface utilisateur. Ceci vous permet d'effectuer la plupart des opérations de manipulation des données traitées de façon automatisée. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Les expressions CLEM se composent de valeurs, de noms de champs, d'opérateurs et de fonctions. En utilisant une syntaxe correcte, vous pouvez créer des opérations de données nombreuses et puissantes. [Pour plus d'informations, reportez-vous à la section CLEM Exemples dans le chapitre 7 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Types de données CLEM

Les types de données CLEM peuvent se composer des éléments suivants :

- Entiers
- Réels
- Caractères
- Chaînes
- Listes
- Champs
- Date/heure

Règles des guillemets

Même si IBM® SPSS® Modeler est flexible lorsque vous déterminez des champs, des valeurs, des chaînes et des paramètres utilisés dans une expression CLEM, les règles générales suivantes fournissent la liste des conseils à suivre lorsque vous créez des expressions :

- Chaînes : utilisez toujours des guillemets doubles lorsque vous écrivez des chaînes, comme "Type 2". Vous pouvez également utiliser des guillemets simples mais cela risque de créer une confusion avec les champs placés entre guillemets.

- Champs : utilisez des guillemets simples, uniquement lorsque cela est rendu nécessaire par la présence d'espaces ou d'autres caractères spéciaux, comme 'Order Number'. les champs placés entre guillemets mais non définis dans l'ensemble de données seront lus comme des chaînes.
- Paramètres : utilisez toujours des guillemets simples lorsque vous employez des paramètres, comme '\$P-threshold'.
- Caractères : utilisez toujours des guillemets simples inversés (`), comme dans stripchar(`d`, "drugA").

Pour plus d'informations, reportez-vous à la section Valeurs et types de données dans le chapitre 7 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*. En outre, ces règles sont détaillées dans les rubriques suivantes.

Entiers

Les entiers sont représentés par une séquence de chiffres décimaux. (Facultatif) Vous pouvez placer un signe moins (-) devant un entier afin d'obtenir un nombre négatif—par exemple, 1234, 999, -77.

Le langage CLEM permet de manipuler les entiers de précision arbitraire. La taille maximale de l'entier varie selon votre plate-forme. Si les valeurs sont trop élevées pour être affichées dans un champ de nombre entier, changez le type de champ en Real pour restaurer la valeur.

Réels

Un nombre *Réel* est un nombre à virgule flottante. Les réels sont représentés par un ou plusieurs chiffres, suivis d'un séparateur décimal, suivi d'un ou de plusieurs autres chiffres. Les réels CLEM sont maintenus en double précision.

(Facultatif) Vous pouvez placer un signe moins (-) devant le réel afin d'obtenir un nombre négatif—par exemple, 1.234, 0.999, -77.001. Utilisez la syntaxe `<nombre> e <exponent>` pour exprimer un nombre réel en notation exponentielle, —par exemple, 1234.0e5, 1.7e-2. Lorsque l'application IBM® SPSS® Modeler lit des chaînes numériques en provenance de fichiers et les convertit automatiquement en nombres, les nombres sans chiffre avant ou après le séparateur décimal sont acceptés—par exemple, 999. ou.11. Cependant, ces formes sont incorrectes dans les expressions CLEM.

Remarque : Lors du référencement de nombres réels dans CLEM, un point doit être utilisé comme séparateur décimal, quels que soient les paramètres pour le flux en cours ou les paramètres régionaux. Par exemple précisez

Na > 0.6

plutôt que :

Na > 0.6

Cela s'applique même lorsque le symbole décimal choisi dans la boîte de dialogue Propriétés du flux est la virgule. En outre, cela est conforme à l'instruction générale selon laquelle la syntaxe de code doit être indépendante des conventions et paramètres régionaux.

Caractères

Des caractères (habituellement désignés par **CHAR**) sont généralement employés avec une expression CLEM pour réaliser des tests sur les chaînes. Ainsi, vous pouvez utiliser la fonction `isuppercode` pour déterminer si le premier caractère d'une chaîne est une majuscule. L'expression CLEM suivante se sert d'un caractère pour indiquer que le test doit être effectué sur le premier caractère de la chaîne :

```
isuppercode(subscrs(1, "MyString"))
```

Pour exprimer le code (et non l'emplacement) d'un caractère particulier dans une expression CLEM, utilisez des guillemets simples inversés : ``<caractère>``— par exemple, ``A``, ``Z``.

Remarque : Il n'existe pas de type de stockage **CHAR** pour les champs. Par conséquent, si un champ est calculé ou rempli avec une expression générant un caractère **CHAR**, ce résultat est converti en chaîne.

Chaînes

En général, vous devez placer les chaînes entre guillemets doubles. Les exemples de chaînes sont "c35product2" et "referrerID". Pour signaler des caractères spéciaux dans une chaîne, utilisez une barre oblique inverse—, par exemple, "\$65443". (Pour indiquer une barre oblique inversée, utilisez deux barres obliques inversées, \.) Vous pouvez placer une chaîne entre guillemets simples mais elle risque alors d'être confondue avec un champ placé entre guillemets ('referrerID'). [Pour plus d'informations, reportez-vous à la section Fonctions sur chaînes dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Listes

Une liste est une séquence d'éléments triés, qui peuvent être de nature différente. Les listes sont placées entre crochets ([]), Les exemples de listes sont [1 2 4 16] et ["abc" "def"]. Les listes ne sont pas utilisées pour les valeurs des champs IBM® SPSS® Modeler. Elles sont utilisées pour fournir des arguments aux fonctions, comme `member` et `oneof`.

Champs

Dans les expressions CLEM, les noms autres que les noms de fonction sont considérés comme des noms de champ. Vous pouvez les écrire aussi simplement que `Power`, `val27`, `state_flag`, etc., mais si le nom commence par un chiffre ou contient des caractères non alphabétiques, comme des espaces (à l'exception du tiret bas), placez le nom entre guillemets simples—par exemple, 'Power Increase', '2nd answer', '#101', '\$P-NextField'.

Remarque : les champs placés entre guillemets mais non définis dans l'ensemble de données seront lus comme des chaînes.

Dates

Les calculs de date reposent sur la date de référence spécifiée dans la boîte de dialogue des propriétés du flux. La date de référence par défaut est le 1er janvier 1900. [Pour plus d'informations, reportez-vous à la section Définition d'options générales pour les flux dans le chapitre 5 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Le langage CLEM prend en charge les formats de date suivants.

Format	Exemples
DDMMYY	150163
MMDDYY	011563
YYMMDD	630115
YYYYMMDD	19630115
YYYYDDD	Année à quatre chiffres, suivie d'une valeur à trois chiffres représentant le jour de l'année— (par exemple, 2000032 correspond au 32e jour de l'année 2000, soit le 1er février 2000.
DAY	Jour de la semaine utilisé dans les paramètres régionaux actuels— (par exemple, Monday, Tuesday, etc. en anglais)
MONTH	Mois utilisé dans les paramètres régionaux actuels—par exemple, January, February,
DD/MM/YY	15/01/63
DD/MM/YYYY	15/01/1963
MM/DD/YY	01/15/63
MM/DD/YYYY	01/15/1963
DD-MM-YY	15-01-63
DD-MM-YYYY	15-01-1963
MM-DD-YY	01-15-63
MM-DD-YYYY	01-15-1963
DD.MM.YY	15.01.63
DD.MM.YYYY	15.01.1963
MM.DD.YY	01.15.63
MM.DD.YYYY	01.15.1963
DD-MON-YY	15-JAN-63, 15-jan-63, 15-Jan-63
DD/MON/YY	15/JAN/63, 15/jan/63, 15/Jan/63
DD.MON.YY	15.JAN.63, 15.jan.63, 15.Jan.63
DD-MON-YYYY	15-JAN-1963, 15-jan-1963, 15-Jan-1963
DD/MON/YYYY	15/JAN/1963, 15/jan/1963, 15/Jan/1963
DD.MON.YYYY	15.JAN.1963, 15.jan.1963, 15.Jan.1963
MON YYYY	Jan 2004

Format	Exemples
q Q YYYY	Date représentée par un chiffre (1–4) correspondant au trimestre, suivi de la lettre <i>Q</i> et d'une année à quatre chiffres— (par exemple, le 25 décembre 2004 est représenté par 4 Q 2004).
ww WK YYYY	Valeur à deux chiffres représentant la semaine de l'année, suivie des lettres <i>SM</i> , puis d'une année à quatre chiffres. La semaine de l'année est calculée en supposant que le premier jour de la semaine est un lundi et que la première semaine comprend au moins un jour.

Temps

Le langage CLEM prend en charge les formats d'heure suivants.

Format	Exemples
HHMMSS	120112, 010101, 221212
HHMM	1223, 0745, 2207
MMSS	5558, 0100
HH:MM:SS	12:01:12, 01:01:01, 22:12:12
HH:MM	12:23, 07:45, 22:07
MM:SS	55:58, 01:00
(H)H:(M)M:(S)S	12:1:12, 1:1:1, 22:12:12
(H)H:(M)M	12:23, 7:45, 22:7
(M)M:(S)S	55:58, 1:0
HH.MM.SS	12.01.12, 01.01.01, 22.12.12
HH.MM	12.23, 07.45, 22.07
MM.SS	55.58, 01.00
(H)H.(M)M.(S)S	12.1.12, 1.1.1, 22.12.12
(H)H.(M)M	12.23, 7.45, 22.7
(M)M.(S)S	55.58, 1.0

Opérateurs CLEM

Les opérateurs suivants sont disponibles.

Opération	Commentaires	Priorité (voir section suivante)
or	Utilisé entre deux expressions CLEM. Renvoie la valeur true (vrai) si l'une des deux, ou les deux expressions présentent la valeur true (vrai).	10
and	Utilisé entre deux expressions CLEM. Renvoie la valeur true (vrai) si les deux expressions présentent la valeur true (vrai).	9

Opération	Commentaires	Priorité (voir section suivante)
=	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 est égal à ITEM2.	7
==	Identique à =.	7
/=	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 <i>n'est pas</i> égal à ITEM2.	7
/==	Identique à /=.	7
>	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 est supérieur à ITEM2.	6
>=	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 est supérieur ou égal à ITEM2.	6
<	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 est inférieur à ITEM2.	6
<=	Utilisé entre deux éléments comparables. Renvoie la valeur true (vrai) si ITEM1 est inférieur ou égal à ITEM2.	6
&&=_0	Utilisé entre deux entiers. Equivaut à l'expression booléenne INT1 && INT2 = 0.	6
&&/=_0	Utilisé entre deux entiers. Equivaut à l'expression booléenne INT1 && INT2 /= 0.	6
+	Additionne deux nombres : NUM1 + NUM2.	5
><	Concatène deux chaînes, par exemple, STRING1 >< STRING2.	5
-	Soustrait un nombre à un autre : NUM1 - NUM2. Peut être également utilisé devant un nombre : - NUM.	5
*	Utilisé pour multiplier deux chiffres : NUM1 * NUM2.	4
&&	Utilisé entre deux entiers. Le résultat de l'opération est l'opérateur sur les bits « ET » des entiers INT1 et INT2.	4
&&~	Utilisé entre deux entiers. Le résultat de l'opération est l'opérateur sur les bits « ET » de l'entier INT1 et le complément sur les bits de l'entier INT2.	4
	Utilisé entre deux entiers. Le résultat de l'opération est l'opérateur sur les bits 'OU inclusif' des entiers INT1 et INT2.	4

Opération	Commentaires	Priorité (voir section suivante)
~~	Utilisé devant un entier. Produit le complément sur les bits de INT.	4
&	Utilisé entre deux entiers. Le résultat de l'opération est l'opérateur sur les bits 'OU exclusif' des entiers INT1 et INT2.	4
INT1 << N	Utilisé entre deux entiers. Produit le motif binaire de l'entier INT décalé vers la gauche de N positions.	4
INT1 >> N	Utilisé entre deux entiers. Produit le motif binaire de l'entier INT décalé vers la droite de N positions.	4
/	Utilisé pour diviser un nombre par un autre : NUM1 / NUM2.	4
**	Utilisé entre deux nombres : BASE ** POWER. Renvoie la valeur de base BASE élevée à la puissance POWER.	3
rem	Utilisé entre deux entiers : INT1 rem INT2. Renvoie le reste, INT1 - (INT1 div INT2) * INT2.	2
div	Utilisé entre deux entiers : INT1 div INT2. Effectue la division des entiers.	2

Priorité des opérateurs

La priorité des opérateurs détermine l'ordre dans lequel les opérations des expressions complexes sont traitées, notamment dans les expressions sans parenthèses avec plusieurs opérateurs infixes. Par exemple,

$3 + 4 * 5$

est analysé en tant que $3 + (4 * 5)$ plutôt que $(3 + 4) * 5$ parce que les priorités relatives indiquent que * doit être analysé avant +. A chaque opérateur du langage CLEM est affectée une valeur de priorité. Plus cette valeur est faible, plus l'opérateur est prioritaire. Il sera donc traité avant ceux ayant un niveau de priorité plus élevé.

Informations de référence sur les fonctions

Les fonctions CLEM suivantes sont disponibles lorsque vous travaillez avec des données dans IBM® SPSS® Modeler. Vous entrez ces fonctions sous forme de code dans un grand nombre de boîtes de dialogue, comme celles des noeuds Calculer et Binariser. Vous pouvez également utiliser

le Générateur de formules pour créer des expressions CLEM valides sans avoir à mémoriser des listes de fonctions ou des noms de champs.

Type de fonction	Description
Informations	Servent à obtenir un aperçu des valeurs de champ. Par exemple, la fonction <code>is_string</code> renvoie une valeur true (vrai) pour tous les enregistrements de type chaîne.
Conversion	Servent à créer des champs ou à convertir un type de stockage. Par exemple, la fonction <code>to_timestamp</code> convertit le champ sélectionné en un horodatage.
Comparaison	Servent à comparer des valeurs de champ, entre elles ou par rapport à la chaîne spécifiée. Par exemple, la fonction <code><=</code> permet de savoir si la valeur d'un champ est inférieure ou égale.
Logique	Servent à exécuter des opérations logiques, comme <code>if</code> , <code>then</code> , <code>else</code> .
Numérique	Servent à effectuer des calculs numériques, comme le calcul du logarithme naturel de valeurs de champ.
Trigonométrique	Servent à effectuer des calculs trigonométriques, comme le calcul du cosinus inverse d'un angle.
Probabilité	Renvoie des probabilités basées sur différentes distributions, comme la probabilité qu'une valeur de la distribution <i>t</i> de Student soit inférieure à une valeur spécifique.
Sur les bits	Servent à manipuler des nombres entiers comme éléments binaires.
Aléatoire	Servent à sélectionner des éléments ou à générer des nombres .
Chaîne	Servent à exécuter un grand nombre d'opérations sur des chaînes, comme <code>stripchar</code> , qui permet de supprimer le caractère spécifié.
SoundEx	Sert à trouver des chaînes lorsque l'orthographe exacte est inconnue, sur la base d'hypothèses phonétiques (modalités de prononciation de certaines lettres).
Date et heure	Servent à exécuter un grand nombre d'opérations sur les champs de date, d'heure et d'horodatage.
Séquence	Servent à obtenir un aperçu de la séquence d'enregistrements d'un ensemble de données ou à exécuter des opérations basées sur cette séquence.
Global	Servent à accéder aux valeurs globales créées par un noeud V. globales (Valeurs globales). Par exemple, <code>@MEAN</code> sert à faire référence à la moyenne de toutes les valeurs d'un champ dans tout l'ensemble de données.
Nul et non renseigné	Servent à atteindre, à marquer et souvent à remplir des blancs définis par l'utilisateur ou des valeurs absentes du système. Par exemple, <code>@BLANK(FIELD)</code> permet d'afficher une valeur true (vrai) pour les enregistrements qui comportent des blancs.
Champs spéciaux	Servent à signaler les champs spéciaux en cours d'examen. Par exemple, <code>@FIELD</code> est utilisé lors du calcul de plusieurs champs.

Conventions pour la description des fonctions

Les conventions suivantes sont utilisées dans ce guide pour faire référence aux éléments d'une fonction.

Convention	Description
<i>BOOL</i>	Un booléen, comme true ou false (vrai, faux).
<i>NUM</i> , <i>NUM1</i> , <i>NUM2</i>	Tout nombre.
<i>REAL</i> , <i>REAL1</i> , <i>REAL2</i>	Nombre réel, comme 1.234 ou -77.01.
<i>INT</i> , <i>INT1</i> , <i>INT2</i>	Un entier, comme 1 ou -77.
<i>CHAR</i>	Code de caractère, comme 'A'.
<i>STRING</i>	Chaîne, comme "referrerID".
<i>LIST</i>	Liste d'éléments, comme ["abc" "def"].
<i>ITEM</i>	Un champ, comme Customer ou extract_concept.
<i>DATE</i>	Champ de date, comme start_date, dont les valeurs sont au format DD-MON-YYYY.
<i>TIME</i>	Champ d'heure, comme power_flux, dont les valeurs sont au format HHMMSS.

Les fonctions couvertes dans ce guide sont répertoriées dans un tableau à trois colonnes : la première colonne indique le nom de la fonction, la deuxième le type de résultat (entier, chaîne, etc.) et la troisième, une description (si elle est disponible). Par exemple, la description suivante est la description d'une fonction rem.

Fonction	Résultat	Description
INT1 rem INT2	<i>Nombre</i>	Renvoie le reste de la division de l'entier <i>INT1</i> par l'entier <i>INT2</i> . Par exemple, $INT1 - (INT1 \text{ div } INT2) * INT2$

Les informations relatives aux conventions d'utilisation, comme la manière de répertorier des éléments ou de spécifier les caractères d'une fonction, sont décrites dans une autre rubrique.

Pour plus d'informations, reportez-vous à la section Types de données CLEM dans [Guide de l'utilisateur de IBM SPSS Modeler 15](#).

Fonctions d'informations

Les fonctions d'informations sont utilisées pour obtenir un aperçu des valeurs d'un champ particulier. Elles sont généralement utilisées pour calculer des champs booléens. Par exemple, vous pouvez utiliser la fonction @BLANK pour créer un champ booléen indiquant les enregistrements dont les valeurs sont vides pour le champ sélectionné. De même, vous pouvez

vérifier le type de stockage d'un champ à l'aide de l'une des fonctions de type de stockage, comme `is_string`.

Fonction	Résultat	Description
@BLANK(FIELD)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements dont les valeurs sont non renseignées, d'après les règles de traitement des blancs définies dans un noeud Typers ou un noeud source en amont (onglet Types). Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
@NULL(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements dont les valeurs ne sont pas définies. Les valeurs non définies sont des valeurs système nulles, affichées dans IBM® SPSS® Modeler sous la forme \$null\$. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
is_date(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type date.
is_datetime(ITEM)	Booléen	Renvoie la valeur vraie (True) pour tous les enregistrements de type date, heure ou horodatage.
is_integer(ITEM)	Booléen	Renvoie la valeur vraie (True) pour tous les enregistrements de type entier.
is_number(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type nombre.
is_real(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type réel.
is_string(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type chaîne.
is_time(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type heure.
is_timestamp(ITEM)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements de type horodatage.

Fonctions de conversion

Les fonctions de conversion permettent de créer des champs et de convertir le type de stockage des champs existants. Par exemple, vous pouvez former de nouvelles chaînes en joignant ou en séparant des chaînes existantes. Pour joindre des chaînes, utilisez l'opérateur `><`. Par exemple, si le champ `Site` a la valeur "BRAMLEY", alors `"xx" >< Site` renvoie "xxBRAMLEY". Le résultat de `><` est toujours une chaîne, même si les arguments n'en sont pas. Par conséquent, si le champ `V1` est 3 et le champ `V2` est 5, alors `V1 >< V2` renvoie "35" (une chaîne pas un chiffre).

Les fonctions de conversion (et toutes les autres fonctions qui nécessitent un type spécifique d'entrée, par exemple une valeur de date ou d'heure) dépendent des formats actuels indiqués dans la boîte de dialogue des options de flux. Par exemple, si vous souhaitez convertir un champ de type chaîne avec des valeurs *Jan 2003*, *Fév 2003*, etc., sélectionnez `MOIS AAAA` comme format de date

par défaut pour le flux. [Pour plus d'informations, reportez-vous à la section Définition d'options générales pour les flux dans le chapitre 5 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Fonction	Résultat	Description
ITEM1 << ITEM2	Chaîne	Concatène les valeurs pour deux champs et renvoie la chaîne résultante sous la forme <i>ITEM1ITEM2</i> .
to_integer(ITEM)	Entier	Convertit le stockage du champ spécifié en un entier.
to_real(ITEM)	Réel	Convertit le stockage du champ spécifié en un réel.
to_number(ITEM)	Nombre	Convertit le stockage du champ spécifié en un nombre.
to_string(ITEM)	Chaîne	Convertit le stockage du champ spécifié en une chaîne.
to_time(ITEM)	Temps	Convertit le stockage du champ spécifié en une heure.
to_date(ITEM)	Date	Convertit le stockage du champ spécifié en une date.
to_timestamp(ITEM)	Horodatage	Convertit le stockage du champ spécifié en un horodatage.
to_datetime(ITEM)	Datetime	Convertit le stockage du champ spécifié en une valeur d'heure, de date ou d'horodatage.
datetime_date(ITEM)	Date	Renvoie la valeur de date pour un <i>nombre</i> , une <i>chaîne</i> , ou un <i>horodatage</i> . Veuillez noter que cette fonction est la seule fonction vous permettant de reconverter un nombre (en secondes) en date. Si ITEM est une chaîne, crée une date en analysant une chaîne au format de date sélectionné. Pour que cette fonction réussisse, vous devez spécifier un format de date correct dans la boîte de dialogue des propriétés du flux. Si ITEM est un nombre, il est interprété comme un nombre de secondes écoulées depuis la date de départ (ou période). Les fractions de jour sont supprimées. Si ITEM est un horodatage, la partie date de cet horodatage est renvoyée. Si ITEM est une date, il est renvoyé sans modification.

Fonctions de comparaison

Les fonctions de comparaison sont utilisées pour comparer des valeurs de champ, entre elles ou par rapport à la chaîne spécifiée. Par exemple, vous pouvez vérifier l'égalité de deux chaînes en utilisant =. Voici un exemple de vérification de l'égalité des chaînes : `Class = "class 1"`.

Lors d'une comparaison de nombres, *supérieur à* signifie que le nombre est plus proche de l'infini positif et *inférieur à* signifie qu'il se rapproche de l'infini négatif. Autrement dit, tous les nombres négatifs sont inférieurs à tous les nombres positifs.

Fonction	Résultat	Description
count_equal(ITEM1, LIST)	Entier	Renvoie le nombre de valeurs, issues d'une liste de champs, égales à <i>ITEM1</i> ou une valeur nulle si <i>ITEM1</i> est nul. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans <i>Guide de l'utilisateur de IBM SPSS Modeler 15</i>.
count_greater_than(ITEM1, LIST)	Entier	Renvoie le nombre de valeurs, issues d'une liste de champs, supérieures à <i>ITEM1</i> ou une valeur nulle si <i>ITEM1</i> est nul.
count_less_than(ITEM1, LIST)	Entier	Renvoie le nombre de valeurs, issues d'une liste de champs, inférieures à <i>ITEM1</i> ou une valeur nulle si <i>ITEM1</i> est nul.
count_not_equal(ITEM1, LIST)	Entier	Renvoie le nombre de valeurs, issues d'une liste de champs, différentes de <i>ITEM1</i> ou une valeur nulle si <i>ITEM1</i> est nul.

Fonction	Résultat	Description
count_nulls(LIST)	Entier	Renvoie le nombre de valeurs nulles issues d'une liste de champs.
count_non_nulls(LIST)	Entier	Renvoie le nombre de valeurs non nulles issues d'une liste de champs.
date_before(DATE1, DATE2)	Booléen	Utilisée pour vérifier l'ordre des valeurs de date. Renvoie une valeur true (vrai) si DATE1 est antérieure à DATE2.
first_index(ITEM, LIST)	Entier	Renvoie l'index du premier champ contenant ITEM depuis une LISTE de champs ou 0 si aucune valeur n'est trouvée. Pris en charge uniquement pour les types de chaînes, d'entiers et de réels. Pour plus d'informations, reportez-vous à la section Utilisation des données à réponses multiples dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
first_non_null(LIST)	Tout	Renvoie la première valeur non nulle de la liste de champs fournie. Tous les types de stockage sont pris en charge.
first_non_null_index(LIST)	Entier	Renvoie l'index du premier champ dans la LISTE spécifiée contenant une valeur non nulle ou 0 si toutes les valeurs sont nulles. Tous les types de stockage sont pris en charge.
ITEM1 = ITEM2	Booléen	Renvoie la valeur true (vrai) pour les enregistrements où ITEM1 est égal à ITEM2.
ITEM1 /= ITEM2	Booléen	Renvoie la valeur true (vrai) si les deux chaînes sont différentes ou renvoie 0 si elles sont identiques.
ITEM1 < ITEM2	Booléen	Renvoie la valeur true (vrai) pour les enregistrements où ITEM1 est inférieur à ITEM2.
ITEM1 <= ITEM2	Booléen	Renvoie la valeur true (vrai) pour les enregistrements où ITEM1 est inférieur ou égal à ITEM2.
ITEM1 > ITEM2	Booléen	Renvoie la valeur true (vrai) pour les enregistrements où ITEM1 est supérieur à ITEM2.
ITEM1 >= ITEM2	Booléen	Renvoie la valeur true (vrai) pour les enregistrements où ITEM1 est supérieur ou égal à ITEM2.
last_index(ITEM, LIST)	Entier	Renvoie l'index du dernier champ contenant ITEM depuis une LISTE de champs ou 0 si aucune valeur n'est trouvée. Pris en charge uniquement pour les types de chaînes, d'entiers et de réels. Pour plus d'informations, reportez-vous à la section Utilisation des données à réponses multiples dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
last_non_null(LIST)	Tout	Renvoie la dernière valeur non nulle de la liste de champs fournie. Tous les types de stockage sont pris en charge.
last_non_null_index(LIST)	Entier	Renvoie l'index du dernier champ dans la LISTE spécifiée contenant une valeur non nulle ou 0 si toutes les valeurs sont nulles. Tous les types de stockage sont pris en charge.
max(ITEM1, ITEM2)	Tout	Renvoie le plus grand des deux éléments ITEM1 et ITEM2.
max_index(LIST)	Entier	Renvoie l'index du champ contenant la valeur maximale d'une liste de champs numériques ou 0 si toutes les valeurs de champ sont nulles. Par exemple, si le troisième champ de la liste contient la valeur maximale, la valeur 3 de l'index est renvoyée. Si plusieurs champs contiennent la valeur maximale, le premier de la liste (le plus à gauche) est renvoyé. Pour plus d'informations, reportez-vous à la section Utilisation des données à réponses multiples dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.

Fonction	Résultat	Description
max_n(LIST)	Nombre	Renvoie la valeur maximale d'une liste de champs numériques ou une valeur nulle si toutes les valeurs de champ sont nulles. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
member(ITEM, LIST)	Booléen	Renvoie la valeur true (vrai) si <i>ITEM</i> est un membre de la liste (<i>LIST</i>) spécifiée. Sinon, renvoie une valeur fausse (False). Il est également possible d'indiquer une liste de noms de champ. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
min(ITEM1, ITEM2)	Tout	Renvoie le plus petit des deux éléments <i>ITEM1</i> et <i>ITEM2</i> .
min_index(LIST)	Entier	Renvoie l'index du champ contenant la valeur minimale d'une liste de champs numériques ou 0 si toutes les valeurs de champ sont nulles. Par exemple, si le troisième champ de la liste contient la valeur minimale, la valeur 3 de l'index est renvoyée. Si plusieurs champs contiennent la valeur minimale, le premier de la liste (le plus à gauche) est renvoyé. Pour plus d'informations, reportez-vous à la section Utilisation des données à réponses multiples dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
min_n(LIST)	Nombre	Renvoie la valeur minimale d'une liste de champs numériques ou une valeur nulle si toutes les valeurs de champ sont nulles.
time_before(TIME1, TIME2)	Booléen	Utilisée pour vérifier l'ordre des valeurs d'heure. Renvoie une valeur true (vrai) si <i>TIME1</i> est antérieure à <i>TIME2</i> .
value_at(INT, LIST)		Renvoie la valeur de décalage INT ou NULL de chaque champ de la liste si le décalage est en-dehors de l'intervalle de valeurs valides (c'est-à-dire, inférieur à 1 ou supérieur au nombre de champs de la liste). Tous les types de stockage sont pris en charge.

Fonctions logiques

Vous pouvez utiliser des expressions CLEM pour effectuer des opérations logiques.

Fonction	Résultat	Description
COND1 and COND2	Booléen	Cette opération est une conjonction logique et renvoie une valeur true (vrai) si <i>COND1</i> et <i>COND2</i> sont toutes les deux vraies. Si <i>COND1</i> est fausse, <i>COND2</i> n'est pas évaluée ; il est alors possible d'avoir des conjonctions où <i>COND1</i> teste d'abord si une opération dans <i>COND2</i> est autorisée. Par exemple, <code>length(Label) >=6 et Label(6) = 'x'</code> .
COND1 or COND2	Booléen	Cette opération est une disjonction logique (inclusive) et renvoie une valeur true (vrai) si l'une des deux conditions <i>COND1</i> or <i>COND2</i> est vraie ou si es deux le sont. Si <i>COND1</i> est vraie, <i>COND2</i> n'est pas évaluée.
not(COND)	Booléen	Cette opération est une négation logique et renvoie une valeur true (vrai) si <i>COND</i> est fausse. Sinon, cette opération renvoie la valeur 0.

Fonction	Résultat	Description
if COND then EXPR1 else EXPR2 endif	Tout	Cette opération est une évaluation conditionnelle. Si <i>COND</i> est vraie, cette opération renvoie le résultat de l'évaluation de l'expression <i>EXPR1</i> . Sinon, elle renvoie le résultat de l'évaluation de l'expression <i>EXPR2</i> .
if COND1 then EXPR1 elseif COND2 then EXPR2 else EXPR_N endif	Tout	Cette opération est une évaluation conditionnelle à plusieurs branches. Si <i>COND1</i> est vraie, cette opération renvoie le résultat de l'évaluation de l'expression <i>EXPR1</i> . En revanche, si <i>COND2</i> est vraie, l'opération renvoie le résultat de l'évaluation de l'expression <i>EXPR2</i> . Sinon, le résultat de l'évaluation de l'expression <i>EXPR_N</i> est renvoyé.

Fonctions numériques

CLEM contient un certain nombre de fonctions numériques courantes.

Fonction	Résultat	Description
-NUM	Nombre	Utilisée pour inverser <i>NUM</i> . Renvoie le même nombre, avec le signe opposé .
NUM1 + NUM2	Nombre	Renvoie la somme de <i>NUM1</i> et <i>NUM2</i> .
code -NUM2	Nombre	Renvoie la valeur <i>NUM2</i> , soustraite de <i>NUM1</i> .
NUM1 * NUM2	Nombre	Renvoie la valeur <i>NUM1</i> multipliée par <i>NUM2</i> .
NUM1 / NUM2	Nombre	Renvoie la valeur <i>NUM1</i> divisée par <i>NUM2</i> .
INT1 div INT2	Nombre	Utilisée pour la division en nombres entiers. Renvoie la valeur de l'entier <i>INT1</i> divisé par l'entier <i>INT2</i> .
INT1 rem INT2	Nombre	Renvoie le reste de la division de l'entier <i>INT1</i> par l'entier <i>INT2</i> . Par exemple, $INT1 - (INT1 \text{ div } INT2) * INT2$
INT1 mod INT2	Nombre	Cette fonction a été remplacée. Utilisez la fonction <i>rem</i> à la place.
BASE ** POWER	Nombre	Renvoie la valeur <i>BASE</i> élevée à la puissance <i>POWER</i> . Les deux valeurs peuvent être tout nombre (en revanche, <i>BASE</i> ne doit pas avoir la valeur zéro si <i>POWER</i> est elle-même égale à une valeur nulle autre que l'entier 0). Si <i>POWER</i> est un entier, le calcul est effectué en multipliant successivement les puissances de <i>BASE</i> . Ainsi, si <i>BASE</i> est un entier, le résultat l'est également. Si <i>POWER</i> est l'entier 0, le résultat est toujours un 1 du même type que <i>BASE</i> . Sinon, si <i>POWER</i> n'est pas un entier, le résultat est calculé selon la formule $\exp(POWER * \log(BASE))$.
abs(NUM)	Nombre	Renvoie la valeur absolue de <i>NUM</i> , qui est toujours un nombre du même type.
exp(NUM)	Réel	Renvoie <i>e</i> élevé à la puissance <i>NUM</i> , <i>e</i> étant la base des logarithmes naturels.
fracof(NUM)	Réel	Renvoie la partie fractionnelle de <i>NUM</i> , définie selon la formule $NUM - \text{intof}(NUM)$.
intof(NUM)	Entier	Tronque son argument pour obtenir un entier. Il renvoie l'entier du même signe que <i>NUM</i> ayant la valeur absolue la plus élevée, comme $\text{abs}(\text{INT}) \leq \text{abs}(\text{NUM})$.
log(NUM)	Réel	Renvoie le logarithme naturel (base <i>e</i>) de <i>NUM</i> , qui ne doit en aucun cas avoir la valeur zéro.

Fonction	Résultat	Description
log10(NUM)	Réel	Renvoie le logarithme décimal de <i>NUM</i> qui ne doit en aucun cas avoir la valeur zéro. Cette fonction est définie selon la formule $\log(\text{NUM}) / \log(10)$.
negate(NUM)	Nombre	Utilisée pour inverser <i>NUM</i> . Renvoie le même nombre, avec le signe opposé.
round(NUM)	Entier	Utilisé pour arrondir <i>NUM</i> à un entier en prenant $\text{intof}(\text{NUM}+0.5)$ si <i>NUM</i> est positif $\text{intof}(\text{NUM}-0.5)$ si <i>NUM</i> est négatif.
sign(NUM)	Nombre	Permet de déterminer le signe de <i>NUM</i> . Cette opération renvoie -1, 0, ou 1 si <i>NUM</i> est un entier. Si <i>NUM</i> est un nombre réel, elle renvoie -1,0, 0,0 ou 1,0 selon que <i>NUM</i> est négatif, positif ou égal à zéro.
sqrt(NUM)	Réel	Renvoie la racine carrée de <i>NUM</i> . <i>NUM</i> doit être positif.
sum_n(LIST)	Nombre	Renvoie la somme des valeurs d'une liste de champs numériques ou une valeur nulle si toutes les valeurs de champ sont nulles. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
mean_n(LIST)	Nombre	Renvoie la valeur moyenne d'une liste de champs numériques ou une valeur nulle si toutes les valeurs de champ sont nulles.
sdev_n(LIST)	Nombre	Renvoie l'écart-type d'une liste de champs numériques ou une valeur nulle si toutes les valeurs de champ sont nulles.

Fonctions trigonométriques

Toutes les fonctions de cette section utilisent un angle comme argument ou en renvoient un comme résultat. Dans les deux cas, les unités d'angle (radians ou degrés) sont contrôlées par les paramètres de l'option de flux appropriée.

Fonction	Résultat	Description
arccos(NUM)	Réel	Calcule le cosinus inverse de l'angle spécifié.
arccosh(NUM)	Réel	Calcule le sinus inverse hyperbolique de l'angle spécifié.
arcsin(NUM)	Réel	Calcule le sinus inverse de l'angle spécifié.
arcsinh(NUM)	Réel	Calcule le sinus inverse hyperbolique de l'angle spécifié.
arctan(NUM)	Réel	Calcule la tangente inverse de l'angle spécifié.
arctan2(NUM_Y, NUM_X)	Réel	Calcule la tangente inverse de $\text{NUM_Y} / \text{NUM_X}$ et utilise les signes des deux nombres pour calculer des informations du cadran. Le résultat est un réel dans l'intervalle $-\pi < \text{ANGLE} \leq \pi$ (radians) – $-180 < \text{ANGLE} \leq 180$ (degrees)
arctanh(NUM)	Réel	Calcule la tangente inverse hyperbolique de l'angle spécifié.
cos(NUM)	Réel	Calcule le cosinus de l'angle spécifié.
cosh(NUM)	Réel	Calcule le cosinus hyperbolique de l'angle spécifié.
pi	Réel	Cette constante est la meilleure approximation réelle de π .
sin(NUM)	Réel	Calcule le sinus de l'angle spécifié.
sinh(NUM)	Réel	Calcule le sinus hyperbolique de l'angle spécifié.
tan(NUM)	Réel	Calcule la tangente de l'angle spécifié.
tanh(NUM)	Réel	Calcule la tangente hyperbolique de l'angle spécifié.

Fonctions de probabilité

Les fonctions de probabilité renvoient des probabilités basées sur différentes distributions, comme la probabilité qu'une valeur de la distribution t de Student soit inférieure à une valeur spécifique.

Fonction	Résultat	Description
<code>cdf_chisq(NUM, DF)</code>	Réel	Renvoie la probabilité qu'une valeur de la distribution du Chi-deux, avec les degrés de liberté indiqués, soit inférieure au nombre spécifié.
<code>cdf_f(NUM, DF1, DF2)</code>	Réel	Renvoie la probabilité qu'une valeur de la distribution F , avec les degrés de liberté $DF1$ et $DF2$ soit inférieure au nombre spécifié.
<code>cdf_normal(NUM, MEAN, STDDEV)</code>	Réel	Renvoie la probabilité qu'une valeur de la distribution normale, avec la moyenne et l'écart-type indiqués, soit inférieure au nombre spécifié.
<code>cdf_t(NUM, DF)</code>	Réel	Renvoie la probabilité qu'une valeur de la distribution t de Student, avec les degrés de liberté indiqués, soit inférieure au nombre spécifié.

Opérations d'entiers sur les bits

Ces fonctions permettent aux entiers d'être manipulés en tant que profils binaires représentant des valeurs de complément à 2, où la position binaire N a la pondération 2^{*N} . Les bits sont numérotés en commençant par 0. Ces opérations se comportent comme si le bit de signe d'un entier était décalé vers la gauche à l'infini. Ainsi, partout au-delà du bit le plus significatif, un entier positif a le bit 0 et un entier négatif a le bit 1.

Remarque : Il n'est pas possible d'appeler des fonctions sur les bits à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
<code>~~ INT1</code>	Entier	Produit le complément sur les bits de l'entier $INT1$. C'est-à-dire qu'il existe un 1 dans le résultat pour chaque position binaire pour laquelle $INT1$ contient un 0. $~~ INT = -(INT + 1)$ est toujours vraie. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
<code>INT1 INT2</code>	Entier	Le résultat de cette opération est l'opérateur sur les bits "inclusif ou" des entiers $INT1$ et $INT2$. Autrement dit, un 1 est inséré dans le résultat pour chaque position binaire où existe un 1 dans $INT1$, dans $INT2$ ou dans les deux entiers.
<code>INT1 /& INT2</code>	Entier	Le résultat de cette opération est l'opérateur sur les bits "exclusif ou" des entiers $INT1$ et $INT2$. Autrement dit, un 1 est inséré dans le résultat pour chaque position binaire où existe un 1 soit dans $INT1$, soit dans $INT2$, mais pas dans les deux.

Fonction	Résultat	Description
<code>INT1 && INT2</code>	<i>Entier</i>	Produit l'opération booléenne « ET » sur les bits des entiers <i>INT1</i> et <i>INT2</i> . Cela signifie qu'un 1 est présent dans le résultat pour chaque position binaire où il existe un 1 dans les deux entiers <i>INT1</i> et <i>INT2</i> .
<code>INT1 &&~ INT2</code>	<i>Entier</i>	Produit le booléen « ET » de l'entier <i>INT1</i> et le complément sur les bits de l'entier <i>INT2</i> . Cela signifie qu'un 1 est présent dans le résultat pour chaque position binaire où il existe un 1 dans <i>INT1</i> et un 0 dans <i>INT2</i> . Cette opération est identique à <code>INT1 && (~INT2)</code> . Elle s'avère utile pour l'élimination des bits d' <i>INT1</i> défini dans <i>INT2</i> .
<code>INT << N</code>	<i>Entier</i>	Produit le motif binaire de l'entier <i>INT1</i> , décalé vers la gauche de <i>N</i> positions. Utilisez une valeur <i>N</i> négative pour produire un décalage vers la droite.
<code>INT >> N</code>	<i>Entier</i>	Produit le motif binaire de l'entier <i>INT1</i> , décalé vers la droite de <i>N</i> positions. Utilisez une valeur <i>N</i> négative pour produire un décalage vers la gauche.
<code>INT1 &&=_0 INT2</code>	<i>Booléen</i>	Equivalut à l'expression booléenne <code>INT1 && INT2 != 0</code> , mais est plus efficace.
<code>INT1 &&/=_0 INT2</code>	<i>Booléen</i>	Equivalut à l'expression booléenne <code>INT1 && INT2 == 0</code> , mais est plus efficace.
<code>integer_bitcount(INT)</code>	<i>Entier</i>	Compte le nombre des bits 1 ou 0 dans la représentation de complément à 2 de l'entier <i>INT</i> . Si <i>INT</i> n'est pas négatif, <i>N</i> est le nombre de bits 1. Si <i>INT</i> est négatif, <i>N</i> correspond au nombre de bits 0. En raison de l'extension de signe, il existe un nombre infini de bits 0 dans un entier positif ou de bits 1 dans un entier négatif. C'est toujours le cas que <code>integer_bitcount(INT) = integer_bitcount(-(INT+1))</code> .
<code>integer_leastbit(INT)</code>	<i>Entier</i>	Renvoie la position binaire <i>N</i> du bit le moins significatif défini dans l'entier <i>INT</i> . <i>N</i> est la puissance de 2 la plus élevée par laquelle il est possible de diviser <i>INT</i> sans reste.
<code>integer_length(INT)</code>	<i>Entier</i>	Renvoie la longueur en bits de l'entier <i>INT</i> , sous forme d'entier de complément à 2. C'est-à-dire, <i>N</i> est le plus petit entier comme <code>INT < (1 << N)</code> if <code>INT >= 0</code> <code>INT >= (-1 << N)</code> if <code>INT < 0</code> . Si <i>INT</i> n'est pas négatif, la représentation d' <i>INT</i> sous forme d'entier non signé nécessite un champ d'au moins <i>N</i> bits. Sinon, un minimum de <i>N+1</i> bits est nécessaire pour représenter <i>INT</i> en tant qu'entier signé, quel que soit son signe.
<code>testbit(INT, N)</code>	<i>Booléen</i>	Teste le bit de position <i>N</i> dans l'entier <i>INT</i> et renvoie l'état du bit <i>N</i> sous forme de valeur booléenne, true (vrai) s'il s'agit de 1 et false (faux) pour 0.

Fonctions aléatoires

Les fonctions suivantes sont utilisées pour sélectionner des éléments ou générer des nombres.

Fonction	Résultat	Description
oneof(LIST)	<i>Tout</i>	Renvoie un élément choisi au hasard dans <i>LIST</i> . Vous devez entrer les éléments de liste au format [ITEM1,ITEM2,...,ITEM_N]. Il est également possible d'indiquer une liste de noms de champ. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans Guide de l'utilisateur de IBM SPSS Modeler 15.
random(NUM)	<i>Nombre</i>	Renvoie un nombre aléatoire distribué uniformément et du même type (<i>INT</i> ou <i>REAL</i>), de 1 à <i>NUM</i> . Si vous utilisez un entier, seuls des entiers sont renvoyés. Si vous utilisez un nombre réel (décimal), des nombres réels sont renvoyés (précision décimale déterminée par les options de flux). Le plus grand nombre aléatoire qui peut être renvoyé par la fonction est égal à <i>NUM</i> .
random0(NUM)	<i>Nombre</i>	Possède les mêmes propriétés que <i>random(NUM)</i> , mais commence à 0. Le plus grand nombre aléatoire renvoyé par la fonction sera toujours différent de <i>X</i> .

Fonctions sur chaînes

Dans le langage CLEM, vous pouvez effectuer les opérations suivantes avec des chaînes :

- Comparer des chaînes
- Créer des chaînes
- Accéder à des caractères

Dans une expression CLEM, une chaîne est une suite de caractères insérée entre guillemets doubles ("string quotes"). Tout caractère alphanumérique unique peut être un caractère (CHAR). Ils sont déclarés dans les expressions CLEM à l'aide de guillemets simples sous la forme de `*caractère*`, comme `z`, `A`, ou `2`. Les caractères hors limites ou les indices négatifs d'une chaîne génèrent un comportement indéfini.

Remarque. Les comparaisons entre les chaînes qui utilisent ou non les répercussions SQL peuvent produire des résultats différents lorsqu'il existe des espaces en fin de chaîne.

Fonction	Résultat	Description
allbutfirst(N, STRING)	<i>Chaîne</i>	Renvoie la chaîne <i>STRING</i> où les <i>N</i> premiers caractères ont été supprimés.
allbutlast(N, STRING)	<i>Chaîne</i>	Renvoie la chaîne <i>STRING</i> où les derniers caractères ont été supprimés.
alphabefore(STRING1, STRING2)	<i>Booléen</i>	Utilisée pour vérifier l'ordre alphabétique des chaînes. Renvoie une valeur true (vrai) si <i>STRING1</i> précède <i>STRING2</i> .
endstring(LENGTH, STRING)	<i>Chaîne</i>	Extrait les <i>N</i> derniers caractères de la chaîne indiquée. Si la longueur de la chaîne est inférieure ou égale à la valeur spécifiée, aucune modification n'a lieu.
hasendstring(STRING, SUBSTRING)	<i>Entier</i>	Cette fonction est la même que <i>isendstring(SUBSTRING, STRING)</i> .

Fonction	Résultat	Description
hasmidstring(String, Substring)	Entier	Cette fonction est la même que ismidstring(Substring, String) (sous-chaîne imbriquée).
hasstartstring(String, Substring)	Entier	Cette fonction est la même que isstartstring(Substring, String).
hassubstring(String, N, Substring)	Entier	Cette fonction est la même que issubstring(Substring, N, String), où la valeur par défaut de <i>N</i> est 1.
count_substring(String, Substring)	Entier	Renvoie le nombre de fois où la sous-chaîne indiquée apparaît dans la chaîne. Par exemple, count_substring("foooo.txt", "oo") renvoie 3.
hassubstring(String, Substring)	Entier	Cette fonction est la même que issubstring(Substring, 1, String), où la valeur par défaut de <i>N</i> est 1.
isalphacode(Char)	Booléen	Renvoie la valeur true (vrai) si <i>CHAR</i> est un caractère dont le code de caractère est une lettre dans la chaîne indiquée (souvent un nom de champ). Sinon, cette fonction renvoie la valeur 0. Par exemple, isalphacode(produce_num(1)).
isendstring(Substring, String)	Entier	Si la chaîne <i>STRING</i> se termine par la sous-chaîne <i>SUBSTRING</i> , cette fonction renvoie l'indice d'entier de <i>SUBSTRING</i> dans <i>STRING</i> . Sinon, cette fonction renvoie la valeur 0.
islowercode(Char)	Booléen	Renvoie la valeur true (vrai) si <i>CHAR</i> est un caractère en minuscule dans la chaîne indiquée (souvent un nom de champ). Sinon, cette fonction renvoie la valeur 0. Par exemple, islowercode(`) et islowercode(country_name(2)) sont toutes deux des expressions valides.
ismidstring(Substring, String)	Entier	Si <i>SUBSTRING</i> est une sous-chaîne de la chaîne <i>STRING</i> , mais qu'elle ne commence pas par le premier caractère de <i>STRING</i> et ne se termine pas par le dernier, cette fonction renvoie l'indice auquel la sous-chaîne commence. Sinon, cette fonction renvoie la valeur 0.
isnumbercode(Char)	Booléen	Renvoie la valeur true (vrai) si <i>CHAR</i> est un caractère dont le code de caractère est un chiffre dans la chaîne indiquée (souvent un nom de champ). Sinon, cette fonction renvoie la valeur 0. Par exemple, isnumbercode(product_id(2)).
isstartstring(Substring, String)	Entier	Si la chaîne <i>STRING</i> commence par la sous-chaîne <i>SUBSTRING</i> , cette fonction renvoie l'indice 1. Sinon, elle renvoie une valeur 0.

Fonction	Résultat	Description
issubstring(SUBSTRING, N, STRING)	Entier	Recherche dans la chaîne <i>STRING</i> , à partir du caractère numéro <i>N</i> , la sous-chaîne <i>SUBSTRING</i> . Si cette sous-chaîne est trouvée, la fonction renvoie l'indice d'entier auquel la sous-chaîne correspondante commence. Sinon cette fonction renvoie une valeur de 0. Si la valeur de <i>N</i> n'est pas fournie, cette fonction prend par défaut la valeur 1.
issubstring(SUBSTRING, STRING)	Entier	Recherche dans la chaîne <i>STRING</i> , à partir du caractère numéro <i>N</i> , la sous-chaîne <i>SUBSTRING</i> . Si cette sous-chaîne est trouvée, la fonction renvoie l'indice d'entier auquel la sous-chaîne correspondante commence. Sinon cette fonction renvoie une valeur de 0. Si la valeur de <i>N</i> n'est pas fournie, cette fonction prend par défaut la valeur 1.
issubstring_count(SUBSTRING, N, STRING):	Entier	Renvoie l'index de la <i>N</i> ème occurrence de la sous-chaîne <i>SUBSTRING</i> au sein de la chaîne <i>STRING</i> indiquée. Si le nombre d'occurrences de la sous-chaîne <i>SUBSTRING</i> est inférieur à <i>N</i> occurrences, 0 est renvoyé.
issubstring_lim(SUBSTRING, N, STARTLIM, ENDLIM, STRING)	Entier	Cette fonction est identique à <i>issubstring</i> , mais la correspondance doit commencer à l'indice <i>STARTLIM</i> ou avant, et doit se terminer à l'indice <i>ENDLIM</i> ou avant. Les contraintes <i>STARTLIM</i> ou <i>ENDLIM</i> peuvent être désactivées en donnant une valeur fausse à l'un des arguments—par exemple, <i>issubstring_lim(SUBSTRING, N, false, false, STRING)</i> est égale à <i>issubstring</i> .
isuppercode(CHAR)	Booléen	Renvoie une valeur true (vrai) si <i>CHAR</i> est un caractère majuscule. Sinon, cette fonction renvoie la valeur 0. Par exemple, <i>isuppercode('')</i> et <i>isuppercode(country_name(2))</i> sont toutes deux des expressions valides.
last(CHAR)	Chaîne	Renvoie le dernier caractère <i>CHAR</i> de la chaîne <i>STRING</i> (qui doit contenir au moins un caractère).
length(STRING)	Entier	Renvoie la longueur de la chaîne <i>STRING</i> , en nombre de caractères.

Fonction	Résultat	Description
locchar(Char, N, STRING)	Entier	Sert à identifier l'emplacement de caractères dans des champs symboliques. Cette fonction recherche le caractère <i>CHAR</i> dans la chaîne <i>STRING</i> , en commençant par le caractère numéro <i>N</i> de <i>STRING</i> . Cette fonction renvoie une valeur indiquant l'emplacement (à partir de <i>N</i>) où se trouve le caractère. Si le caractère est introuvable, la fonction renvoie la valeur 0. Si la fonction a un décalage (<i>N</i>) incorrect (par exemple, décalage dépassant la longueur de la chaîne), elle renvoie \$null\$. Par exemple, locchar(`n`, 2, web_page) recherche le caractère `n` dans le champ <i>web_page</i> en commençant par le deuxième caractère de la valeur du champ. <i>Remarque</i> : Veuillez à encadrer le caractère spécifié de guillemets simples inversés.
locchar_back(Char, N, STRING)	Entier	Semblable à locchar, cette fonction effectue la recherche vers l'arrière, en commençant par le caractère numéro <i>N</i> . Par exemple, locchar_back(`n`, 9, web_page) effectue une recherche dans le champ <i>web_page</i> en commençant par le neuvième caractère, puis en repartant vers le début de la chaîne. Si cette fonction a un décalage incorrect (par exemple, décalage dépassant la longueur de la chaîne), elle renvoie \$null\$. Pour plus d'efficacité, employez locchar_back en association avec la fonction length(<field>) afin d'utiliser la longueur de la valeur actuelle du champ de manière dynamique Par exemple, locchar_back(`n`, (length(web_page)), web_page)
lowertoupper(Char) lowertoupper(STRING)	CHAR ou Chaîne	L'entrée peut être soit une chaîne soit un caractère, qui est utilisé dans cette fonction pour renvoyer un nouvel élément de même type, avec les caractères minuscules convertis dans leurs équivalents majuscules. Par exemple, lowertoupper(`a`), lowertoupper("My string"), et lowertoupper(field_name(2)) sont toutes des expressions valides.
matches	Booléen	Renvoie la valeur True (vrai) si une chaîne correspond à un patron spécifié. Le patron doit être un littéral de chaîne ; il ne peut pas s'agir d'un nom de champ contenant un patron. Il est possible d'inclure un point d'interrogation (?) dans le patron pour obtenir une correspondance exacte avec un caractère ; un astérisque (*) implique une correspondance avec zéro ou plusieurs caractères. Pour obtenir une correspondance avec un point d'interrogation ou un astérisque de littéral (plutôt que de les utiliser comme caractères génériques), il est

Fonction	Résultat	Description
		possible d'utiliser une barre oblique inverse (<code>\</code>) en tant que caractère d'échappement.
<code>replace(SUBSTRING, NEWSUBSTRING, STRING)</code>	<i>Chaîne</i>	Dans la chaîne <i>STRING</i> indiquée, remplace toutes les instances de la sous-chaîne <i>SUBSTRING</i> par la nouvelle sous-chaîne <i>NEWSUBSTRING</i> .
<code>replicate(COUNT, STRING)</code>	<i>Chaîne</i>	Renvoie une chaîne qui se compose de la chaîne d'origine copiée le nombre de fois indiqué.
<code>stripchar(CHAR,STRING)</code>	<i>Chaîne</i>	Cette fonction permet de supprimer des caractères d'une chaîne ou d'un champ. Vous pouvez l'utiliser, par exemple, pour supprimer de vos données des symboles de devises afin d'obtenir un simple nombre ou un nom. Par exemple, la syntaxe <code>stripchar('\$', 'Cost')</code> renvoie un nouveau champ où toutes les valeurs apparaissent sans le symbole dollar. <i>Remarque</i> : Veillez à encadrer le caractère spécifié de guillemets simples inversés.
<code>skipchar(CHAR, N, STRING)</code>	<i>Entier</i>	Recherche tout autre caractère que <i>CHAR</i> dans la chaîne <i>STRING</i> , en commençant par le caractère numéro <i>N</i> . Cette fonction renvoie un indice d'entier indiquant l'endroit auquel le caractère a été trouvé. Si tous les caractères qui suivent le caractère numéro <i>N</i> sont des caractères <i>CHAR</i> , la fonction renvoie 0. Si cette fonction a un décalage incorrect (par exemple, décalage dépassant la longueur de la chaîne), elle renvoie <code>\$null\$</code> . <code>locchar</code> est souvent utilisé en association avec les fonctions <code>skipchar</code> afin de déterminer la valeur de <i>N</i> (point de la chaîne à partir duquel commence la recherche) Par exemple, <code>skipchar('s', (locchar('s', 1, "MyString")), "MyString")</code>
<code>skipchar_back(CHAR, N, STRING)</code>	<i>Entier</i>	Semblable à <code>skipchar</code> , cette fonction effectue la recherche vers l'arrière , en commençant par le caractère numéro <i>N</i> .
<code>startstring(LENGTH, STRING)</code>	<i>Chaîne</i>	Extrait les <i>N</i> premiers caractères de la chaîne indiquée. Si la longueur de la chaîne est inférieure ou égale à la valeur spécifiée, aucune modification n'a lieu.
<code>strmember(CHAR, STRING)</code>	<i>Entier</i>	Semblable à <code>locchar(CHAR, 1, STRING)</code> . Renvoie un indice entier qui indique l'endroit où <i>CHAR</i> s'est produit pour la première fois, ou 0. Si la fonction a un décalage non valide (par exemple, un décalage au-delà de la longueur de la chaîne), cette fonction renvoie <code>\$null\$</code> .
<code>subscrs(N, STRING)</code>	<i>CHAR</i>	Renvoie le caractère (<i>CHAR</i>) numéro <i>N</i> de la chaîne de saisie <i>STRING</i> . Cette fonction peut également être écrite en sténographie, <code>STRING(N)</code> . Par exemple, <code>lowertoupper("name"(1))</code> est une expression valide.

Fonction	Résultat	Description
substring(N, LEN, STRING)	Chaîne	Renvoie la sous-chaîne <i>SUBSTRING</i> , constituée de <i>LEN</i> caractères de la chaîne <i>STRING</i> , à partir du caractère situé à l'indice <i>N</i> .
substring_between(N1, N2, STRING)	Chaîne	Renvoie la sous-chaîne de la chaîne <i>STRING</i> qui commence à l'indice <i>N1</i> et se termine à l'indice <i>N2</i> .
trim(STRING)	Chaîne	Supprime les espaces blancs situés au début et à la fin de la chaîne indiquée.
trim_start(STRING)	Chaîne	Supprime les espaces blancs situés au début de la chaîne indiquée.
trimend(STRING)	Chaîne	Supprime les espaces blancs situés à la fin de la chaîne indiquée.
unicode_char(NUM)	CHAR	Renvoie le caractère avec la valeur Unicode <i>NUM</i> .
unicode_value(CHAR)	NUM	Renvoie la valeur Unicode de <i>CHAR</i> .
uppertolower(CHAR) uppertolower (STRING)	CHAR ou Chaîne	L'entrée peut être soit une chaîne soit un caractère, qui est utilisé dans cette fonction pour renvoyer un nouvel élément de même type avec les caractères en majuscule convertis dans leurs équivalents en minuscule. <i>Remarque</i> : n'oubliez pas d'encadrer les chaînes spécifiées de guillemets doubles et les caractères de guillemets simples inversés. Les noms de champ simples ne doivent pas contenir de guillemets.

Fonctions SoundEx

SoundEx est une méthode qui permet de trouver des chaînes lorsque le son est connu mais pas l'orthographe exacte. Développée en 1918, elle recherche des mots présentant un son similaire, sur la base d'hypothèses phonétiques (modalités de prononciation de certaines lettres). Il est possible, par exemple, de l'utiliser pour rechercher des noms dans une base de données où l'orthographe et la prononciation de noms similaires peuvent varier. L'algorithme SoundEx de base est décrit dans diverses sources et, malgré un certain nombre de limitations connues (par exemple, les combinaisons de lettres de début telles que ph et f n'entraînent pas de correspondance même si le son est identique), il est pris en charge d'une certaine manière par la plupart des bases de données.

Fonction	Résultat	Description
soundex(STRING)	Entier	Renvoie le code SoundEx à quatre caractères pour la chaîne <i>STRING</i> indiquée.
soundex_difference(STRING1, STRING2)	Entier	Renvoie un entier entre 0 et 4 qui indique le nombre de caractères identiques dans le codage SoundEx pour les deux chaînes (0 = absence de similarité, 4 = similarité élevée ou chaînes identiques).

Fonctions date et heure

CLEM comprend une série de fonctions permettant de manipuler des champs avec un stockage datetime contenant des variables de chaîne représentant des dates et des heures. Les formats de date et d'heure utilisés sont propres à chaque flux et sont définis dans la boîte de dialogue Propriétés du flux. Les fonctions de date et d'heure analysent les chaînes de date et d'heure en fonction du format sélectionné.

Lorsque vous indiquez dans une date une année sur deux chiffres seulement (c'est-à-dire sans préciser le siècle), IBM® SPSS® Modeler utilise le siècle par défaut défini dans la boîte de dialogue des propriétés du flux.

Remarque : Il n'est pas possible d'appeler des fonctions de date et d'heure à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
@TODAY	Chaîne	Si vous sélectionnez Passer jours/minutes dans la boîte de dialogue des propriétés du flux, cette fonction renvoie la date actuelle en tant que chaîne sous le format de date actuel. Si vous utilisez un format de date sur deux chiffres sans sélectionner l'option Passer jours/minutes, la fonction renvoie la valeur \$null\$ sur le serveur actuel. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
to_time(ITEM)	Temps	Convertit le stockage du champ spécifié en une heure.
to_date(ITEM)	Date	Convertit le stockage du champ spécifié en une date.
to_timestamp(ITEM)	Horodatage	Convertit le stockage du champ spécifié en un horodatage.
to_datetime(ITEM)	Datetime	Convertit le stockage du champ spécifié en une valeur d'heure, de date ou d'horodatage.
datetime_date(ITEM)	Date	Renvoie la valeur de date pour un <i>nombre</i> , une <i>chaîne</i> , ou un <i>horodatage</i> . Veuillez noter que cette fonction est la seule fonction vous permettant de reconverter un nombre (en secondes) en date. Si ITEM est une chaîne, crée une date en analysant une chaîne au format de date sélectionné. Pour que cette fonction réussisse, vous devez spécifier un format de date correct dans la boîte de dialogue des propriétés du flux. Si ITEM est un nombre, il est interprété comme un nombre de secondes écoulées depuis la date de départ (ou période). Les fractions de jour sont supprimées. Si ITEM est un horodatage, la partie date de cet horodatage est renvoyée. Si ITEM est une date, il est renvoyé sans modification.
date_before(DATE1, DATE2)	Booléen	Renvoie une valeur true (vrai) si DATE1 représente une date ou un horodatage antérieur à DATE2. Sinon, cette fonction renvoie la valeur 0.
date_days_difference(DATE1, DATE2)	Entier	Renvoie la durée en jours depuis la date ou l'horodatage qui sépare DATE1 de DATE2, sous forme d'entier. Si DATE2 est antérieure à DATE1, cette fonction renvoie un nombre négatif.

Fonction	Résultat	Description
date_in_days(<i>DATE</i>)	<i>Entier</i>	Renvoie la durée en jours qui sépare la date ou l'horodatage de référence de l'élément <i>DATE</i> indiqué, sous la forme d'un entier. Si <i>DATE</i> est antérieure à la date de référence, cette fonction renvoie un nombre négatif. Vous devez inclure une date réelle pour que le calcul fonctionne correctement. Par exemple, vous ne devez pas entrer la date : 29 février 2001. En effet, 2001 n'est pas une année bissextile et cette date n'existe pas.
date_in_months(<i>DATE</i>)	<i>Réel</i>	Renvoie la durée en mois qui sépare la date ou l'horodatage de référence de l'élément <i>DATE</i> indiqué, sous la forme d'un nombre réel. Ce chiffre est approximatif et utilise une base de 30.4375 jours par mois. Si <i>DATE</i> est antérieure à la date de référence, cette fonction renvoie un nombre. Vous devez inclure une date réelle pour que le calcul fonctionne correctement. Par exemple, vous ne devez pas entrer la date : 29 février 2001. En effet, 2001 n'est pas une année bissextile et cette date n'existe pas.
date_in_weeks(<i>DATE</i>)	<i>Réel</i>	Renvoie la durée en semaines qui sépare la date ou l'horodatage de référence de l'élément <i>DATE</i> indiqué, sous la forme d'un nombre réel. La semaine de base utilisée comprend 7,0 jours. Si <i>DATE</i> est antérieure à la date de référence, cette fonction renvoie un nombre négatif. Vous devez inclure une date réelle pour que le calcul fonctionne correctement. Par exemple, vous ne devez pas entrer la date : 29 février 2001. En effet, 2001 n'est pas une année bissextile et cette date n'existe pas.
date_in_years(<i>DATE</i>)	<i>Réel</i>	Renvoie la durée en années qui sépare la date ou l'horodatage de référence de l'élément <i>DATE</i> indiqué, sous la forme d'un nombre réel. Ce chiffre est approximatif et utilise une base de 365.25 jours. Si <i>DATE</i> est antérieure à la date de référence, cette fonction renvoie un nombre négatif. Vous devez inclure une date réelle pour que le calcul fonctionne correctement. Par exemple, vous ne devez pas entrer la date : 29 février 2001. En effet, 2001 n'est pas une année bissextile et cette date n'existe pas.
date_months_difference(<i>DATE1</i> , <i>DATE2</i>)	<i>Réel</i>	Renvoie la durée en mois depuis la date ou l'horodatage qui sépare <i>DATE1</i> de <i>DATE2</i> , sous forme d'un nombre réel. Ce chiffre est approximatif et utilise une base de 30.4375 jours par mois. Si <i>DATE2</i> est antérieure à <i>DATE1</i> , cette fonction renvoie un nombre négatif.
datetime_date(<i>YEAR</i> , <i>MONTH</i> , <i>DAY</i>)	<i>Date</i>	Renvoie la valeur de date pour les éléments <i>YEAR</i> , <i>MONTH</i> et <i>DAY</i> (année, mois, jour) indiqués. Ces arguments doivent être des entiers.
datetime_day(<i>DATE</i>)	<i>Entier</i>	Renvoie le jour du mois à partir de la valeur <i>DATE</i> ou d'un horodatage. Le résultat est un entier compris entre 1 et 31.
datetime_day_name(<i>DAY</i>)	<i>Chaîne</i>	Renvoie le nom complet de l'élément <i>DAY</i> (jour) indiqué. L'argument doit être un entier compris entre 1 (dimanche) et 7 (samedi).

Fonction	Résultat	Description
datetime_hour(TIME)	Entier	Renvoie l'heure à partir de la valeur <i>TIME</i> ou d'un horodatage. Le résultat est un entier compris entre 0 et 23.
datetime_in_seconds(TIME)	Réel	Renvoie la partie des secondes stockée dans <i>TIME</i> .
datetime_in_seconds(DATE), datetime_in_seconds(DATE-TIME)	Réel	Renvoie le chiffre cumulé, converti en secondes, de la différence entre la date <i>DATE</i> ou <i>DATETIME</i> actuelle et la date de référence (1900-01-01).
datetime_minute(TIME)	Entier	Renvoie le nombre de minutes, à partir de <i>TIME</i> ou d'un horodatage. Le résultat est un entier compris entre 0 et 59.
datetime_month(DATE)	Entier	Renvoie le mois, à partir de <i>DATE</i> ou d'un horodatage. Le résultat est un entier compris entre 1 et 12.
datetime_month_name(MONTH)	Chaîne	Renvoie le nom complet de l'élément <i>MONTH</i> (mois) indiqué. L'argument doit être un entier compris entre 1 et 12.
datetime_now	Horodatage	Renvoie l'heure actuelle sous la forme d'un horodatage.
datetime_second(TIME)	Entier	Renvoie le nombre de secondes, à partir de <i>TIME</i> ou d'un horodatage. Le résultat est un entier compris entre 0 et 59.
datetime_day_short_name(DAY)	Chaîne	Renvoie le nom abrégé de l'élément <i>DAY</i> (jour) indiqué. L'argument doit être un entier compris entre 1 (dimanche) et 7 (samedi).
datetime_month_short_name(MONTH)	Chaîne	Renvoie le nom abrégé de l'élément <i>MONTH</i> (mois) indiqué. L'argument doit être un entier compris entre 1 et 12.
datetime_time(HOUR, MINUTE, SECOND)	Temps	Renvoie la valeur d'heure pour les éléments <i>HOUR</i> , <i>MINUTE</i> et <i>SECOND</i> (heures, minutes, secondes) indiqués. Ces arguments doivent être des entiers.
datetime_time(ITEM)	Temps	Renvoie la valeur d'heure de l'élément <i>ITEM</i> indiqué.
datetime_timestamp(YEAR, MONTH, DAY, HOUR, MINUTE, SECOND)	Horodatage	Renvoie la valeur d'horodatage pour les éléments <i>YEAR</i> , <i>MONTH</i> , <i>DAY</i> , <i>HOUR</i> , <i>MINUTE</i> et <i>SECOND</i> (année, mois, jour, heures, minutes, secondes) indiqués.
datetime_timestamp(DATE, TIME)	Horodatage	Renvoie la valeur d'horodatage des éléments <i>DATE</i> et <i>TIME</i> (date, heure) indiqués.
datetime_timestamp (NUMBER)	Horodatage	Renvoie la valeur d'horodatage du nombre de secondes indiqué.
datetime_weekday(DATE)	Entier	Renvoie le jour de la semaine à partir de la valeur <i>DATE</i> ou d'un horodatage.
datetime_year(DATE)	Entier	Renvoie l'année, à partir de <i>DATE</i> ou d'un horodatage. Le résultat est un entier (comme 2002).
date_weeks_difference (DATE1, DATE2)	Réel	Renvoie la durée en semaines depuis la date ou l'horodatage qui sépare <i>DATE1</i> de <i>DATE2</i> , sous forme d'un nombre réel. La semaine de base utilisée comprend 7,0 jours. Si <i>DATE2</i> est antérieure à <i>DATE1</i> , cette fonction renvoie un nombre négatif.
date_years_difference (DATE1, DATE2)	Réel	Renvoie la durée en années depuis la date ou l'horodatage qui sépare <i>DATE1</i> de <i>DATE2</i> , sous forme d'un nombre réel. Ce chiffre est approximatif et utilise une base de 365.25 jours. Si <i>DATE2</i> est antérieure à <i>DATE1</i> , cette fonction renvoie un nombre négatif.

Fonction	Résultat	Description
time_before(TIME1, TIME2)	Booléen	Renvoie une valeur true (vrai) si <i>TIME1</i> représente une heure ou un horodatage antérieur à <i>TIME2</i> . Sinon, cette fonction renvoie la valeur 0.
time_hours_difference (TIME1, TIME2)	Réel	Renvoie la durée en heures qui sépare les heures ou les horodatages représentés par <i>TIME1</i> et <i>TIME2</i> , sous la forme d'un nombre réel. Si vous sélectionnez l'option Passer jours/minutes dans la boîte de dialogue des propriétés du flux, une valeur plus élevée de <i>TIME1</i> fait référence au jour précédent. Si vous ne sélectionnez pas cette option, une valeur plus élevée de <i>TIME1</i> entraîne une valeur négative.
time_in_hours(TIME)	Réel	Renvoie le nombre d'heures représenté par l'heure <i>TIME</i> , sous la forme d'un nombre réel. Par exemple, pour le format d'heure HHMM, l'expression time_in_hours('0130') renvoie la valeur 1,5. <i>TIME</i> peut représenter une heure ou un horodatage.
time_in_mins(TIME)	Réel	Renvoie le nombre de minutes représenté par l'heure <i>TIME</i> , sous la forme d'un nombre réel. <i>TIME</i> peut représenter une heure ou un horodatage.
time_in_secs(TIME)	Entier	Renvoie le nombre de secondes représenté par l'heure <i>TIME</i> , sous la forme d'un entier. <i>TIME</i> peut représenter une heure ou un horodatage.
time_mins_difference(TIME1, TIME2)	Réel	Renvoie la durée en minutes qui sépare les heures ou les horodatages représentés par <i>TIME1</i> et <i>TIME2</i> , sous la forme d'un nombre réel. Si vous sélectionnez l'option Passer jours/minutes dans la boîte de dialogue des propriétés du flux, une valeur plus élevée de <i>TIME1</i> est utilisée pour faire référence au jour précédent (ou à l'heure précédente, si les minutes et les secondes sont uniquement spécifiées dans le format actuel). Si vous ne sélectionnez pas cette option, une valeur plus élevée de <i>TIME1</i> entraîne une valeur négative.
time_secs_difference(TIME1, TIME2)	Entier	Renvoie la durée en secondes qui sépare l'heure ou l'horodatage représenté par <i>TIME1</i> de celle représentée par <i>TIME2</i> , sous la forme d'un entier. Si vous sélectionnez l'option Passer jours/minutes dans la boîte de dialogue des propriétés du flux, une valeur plus élevée de <i>TIME1</i> est utilisée pour faire référence au jour précédent (ou à l'heure précédente, si les minutes et les secondes sont uniquement spécifiées dans le format actuel). Si vous ne sélectionnez pas cette option, une valeur plus élevée de <i>TIME1</i> entraîne une valeur négative.

Conversion des valeurs date et heure.

Notez que les fonctions de conversion (et toutes les autres fonctions qui nécessitent un type spécifique d'entrée, par exemple une valeur de date ou d'heure) dépendent des formats actuels indiqués dans la boîte de dialogue des options de flux. Par exemple, si vous avez un champ *DATE* de type chaîne avec les valeurs *Jan 2003*, *Fev 2003*, etc., vous pouvez le convertir en stockage de date sous le format suivant :

to_date(DATE)

Pour que cette conversion fonctionne, sélectionnez le format de date correspondant MON YYYY comme format de date par défaut pour le flux. [Pour plus d'informations, reportez-vous à la section Définition d'options générales pour les flux dans le chapitre 5 dans *Guide de l'utilisateur de IBM SPSS Modeler 15*.](#)

Pour un exemple de conversion de valeurs de chaînes en dates à l'aide d'un noeud Remplacer, consultez le flux *broadband_create_models.str*, installé dans le dossier *\Demos* du sous-dossier *streams*. [Pour plus d'informations, reportez-vous à la section Préviation avec le noeud Séries temporelles dans le chapitre 14 dans *Guide d'applications de IBM SPSS Modeler 15*.](#)

Dates stockées sous forme de nombres. Veuillez noter que *DATE* dans l'exemple précédent est le nom d'un champ, alors que *to_date* est une fonction CLEM. Si vous avez des dates stockées sous la forme de nombres, vous pouvez les convertir en utilisant la fonction *datetime_date* où le nombre est interprété comme un nombre de secondes écoulées depuis la date de départ (ou période).

```
datetime_date(DATE)
```

En convertissant une date en un nombre de secondes (et vice-versa), vous pouvez effectuer des calculs tels que le calcul de la date actuelle plus ou moins un certain nombre de jours, par exemple :

```
datetime_date((date_in_days(DATE)-7)*60*60*24)
```

Fonctions séquentielles

Dans le cadre de certaines opérations, la séquence des événements est importante. Cette application vous permet de travailler avec les séquences d'enregistrements suivantes :

- Séquences et séries temporelles
- Fonctions séquentielles
- Indexation des enregistrements
- Moyenne, somme et valeurs comparatives
- Contrôle des changements - Différenciation
- @SINCE
- Valeurs de décalage
- Outils séquentiels supplémentaires

Pour nombre d'applications, chaque enregistrement circulant dans un flux peut être considéré comme un cas particulier, indépendant de tous les autres. Auxquels cas, l'ordre des enregistrements importe généralement peu.

Toutefois, pour certaines catégories de problèmes, la séquence des enregistrements est importante. Il s'agit en général de situations de séries temporelles dans lesquelles la séquence des enregistrements représente une séquence ordonnée d'événements ou d'occurrences. Chaque enregistrement représente un instantané d'un moment particulier. Cependant, une grande partie des informations les plus complexes peut être contenue non pas dans des valeurs instantanées, mais dans la façon dont de telles valeurs évoluent et se comportent dans le temps.

Bien entendu, le paramètre pertinent peut être autre que le temps. Par exemple, les enregistrements peuvent représenter des analyses effectuées à distance sur une ligne, mais les mêmes principes s'appliquent.

Les fonctions séquentielles et les fonctions spéciales sont reconnaissables aux caractéristiques suivantes.

- Elles sont toutes précédées du préfixe @.
- Leur nom apparaît en majuscules.

Les fonctions séquentielles font référence à l'enregistrement actuellement traité par un noeud, aux enregistrements qui sont déjà passés dans un noeud et même, dans un cas, aux enregistrements qui ne sont pas encore passés dans un noeud. Les fonctions séquentielles peuvent être librement mélangées avec d'autres composants d'expressions CLEM, bien que certains aient des restrictions quant à leurs arguments.

Exemples

Vous pouvez avoir besoin de connaître le temps écoulé depuis un certain événement ou une condition vraie. Pour cela, utilisez la fonction @SINCE. Par exemple :

```
@SINCE(Income > Outgoings)
```

Cette fonction renvoie le décalage du dernier enregistrement où cette condition était vraie, c'est-à-dire le nombre d'enregistrements précédant celui où la condition était vraie. Si la condition n'a jamais été vraie, @SINCE renvoie @INDEX + 1.

Vous pouvez également devoir faire référence à une valeur de l'enregistrement en cours dans l'expression utilisée par @SINCE. Pour cela, utilisez la fonction @THIS, qui spécifie qu'un nom de champ s'applique toujours à l'enregistrement en cours. Pour déterminer le décalage du dernier enregistrement dont la valeur de champ Concentration est plus de deux fois supérieure à celle de l'enregistrement en cours, vous pouvez utiliser :

```
@SINCE(Concentration > 2 * @THIS(Concentration))
```

Dans certains cas, la condition attribuée à @SINCE est par définition vraie pour l'enregistrement en cours. Par exemple :

```
@SINCE(ID == @THIS(ID))
```

Pour cette raison, @SINCE n'évalue pas la condition de l'enregistrement en cours. Utilisez une fonction identique, @SINCE0, pour évaluer la condition de l'enregistrement en cours et des précédents. Si la condition est vraie dans l'enregistrement en cours, @SINCE0 renvoie 0.

Remarque : Il n'est pas possible d'appeler des fonctions @ à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
MEAN(FIELD)	Réel	Renvoie la moyenne des valeurs du ou des champs (<i>FIELD</i> ou <i>FIELDS</i>) spécifiés.
@MEAN(FIELD, EXPR)	Réel	Renvoie la moyenne des valeurs du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, y compris l'enregistrement actuel. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la moyenne des enregistrements reçus est renvoyée. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
@MEAN(FIELD, EXPR, INT)	Réel	Renvoie la moyenne des valeurs du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, y compris l'enregistrement actuel. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la moyenne des enregistrements reçus est renvoyée. <i>INT</i> indique le nombre maximal de valeurs à prendre en compte. Cette méthode est beaucoup plus efficace que si vous limitez à deux arguments.
@DIFF1(FIELD)	Réel	Renvoie le premier différentiel du champ <i>FIELD1</i> . La forme à un seul argument renvoie simplement la différence entre la valeur de champ actuelle et la valeur de champ précédente. Renvoie 0 si aucun enregistrement précédent pertinent n'existe.
@DIFF1(FIELD1, FIELD2)	Réel	La forme à deux arguments renvoie le premier différentiel du champ <i>FIELD1</i> par rapport au champ <i>FIELD2</i> . Renvoie 0 si aucun enregistrement précédent pertinent n'existe.
@DIFF2(FIELD)	Réel	Renvoie le second différentiel du champ <i>FIELD1</i> . La forme à un seul argument renvoie simplement la différence entre la valeur de champ actuelle et la valeur de champ précédente. Renvoie 0 si aucun enregistrement précédent pertinent n'existe.
@DIFF2(FIELD1, FIELD2)	Réel	La forme à deux arguments renvoie le premier différentiel du champ <i>FIELD1</i> par rapport au champ <i>FIELD2</i> . Renvoie 0 si aucun enregistrement précédent pertinent n'existe.

Fonction	Résultat	Description
@INDEX	Entier	Renvoie l'index de l'enregistrement actuel. Les indices sont affectés à chaque enregistrement lors de son arrivée sur le noeud actuel. Le premier enregistrement reçoit l'index 1 et la valeur d'index augmente de 1 à chacun des enregistrements suivants.
@LAST_NON_BLANK(FIELD)	Tout	Renvoie la dernière valeur du champ <i>FIELD</i> qui était renseigné, tel que défini dans un noeud <i>Type</i> ou source en amont. S'il n'existe aucune valeur renseignée pour le champ <i>FIELD</i> dans les enregistrements lus jusqu'à présent, la fonction renvoie \$null\$. Les valeurs non renseignées, également appelées valeurs manquantes utilisateur, peuvent être définies séparément pour chaque champ.
@MAX(FIELD)	Nombre	Renvoie la valeur maximale du champ <i>FIELD</i> spécifié.
@MAX(FIELD, EXPR)	Nombre	Renvoie la valeur maximale du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus jusqu'à présent, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression dont le résultat est un entier supérieur à 0.
@MAX(FIELD, EXPR, INT)	Nombre	Renvoie la valeur maximale du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus jusqu'à présent, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la valeur minimale des enregistrements reçus est renvoyée. <i>INT</i> indique le nombre maximal de valeurs à prendre en compte. Cette méthode est beaucoup plus efficace que si vous vous limitez à deux arguments.
@MIN(FIELD)	Nombre	Renvoie la valeur minimale du champ <i>FIELD</i> spécifié.
@MIN(FIELD, EXPR)	Nombre	Renvoie la valeur minimale du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus jusqu'à présent, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression dont le résultat est un entier supérieur à 0.
@MIN(FIELD, EXPR, INT)	Nombre	Renvoie la valeur minimale du champ <i>FIELD</i> correspondant aux <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus jusqu'à présent, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la valeur minimale des enregistrements reçus est renvoyée. <i>INT</i> indique le nombre maximal de valeurs à prendre en compte. Cette méthode est beaucoup plus efficace que si vous vous limitez à deux arguments.

Fonction	Résultat	Description
@OFFSET(FIELD, EXPR)	Tout	Renvoie la valeur de <i>FIELD</i> pour l'enregistrement décalé de la valeur <i>EXPR</i> par rapport à l'enregistrement actuel. Un décalage positif permet d'accéder à un enregistrement qui a déjà eu lieu, alors qu'un décalage négatif indique un enregistrement anticipé. Par exemple, @OFFSET(Status, 1) renvoie la valeur du champ Status de l'enregistrement précédent, alors que @OFFSET(Status, -4) « anticipe » quatre enregistrements dans la séquence (enregistrements qui ne sont pas encore passés par ce noeud) pour obtenir la valeur. <i>Veillez noter qu'un décalage négatif (fonction d'anticipation) doit être spécifié comme constante.</i> Pour les décalages positifs uniquement, <i>EXPR</i> peut également être une expression CLEM arbitraire, évaluée pour que l'enregistrement actuel calcule le décalage. Dans ce cas, l'utilisation de la version à trois arguments de cette fonction devrait améliorer les performances (consultez la fonction suivante). Si le résultat de cette expression n'est pas un entier non négatif, une erreur est générée. Il est en effet interdit de calculer des décalages anticipés. <i>Remarque :</i> Une fonction @OFFSET autoréférentielle ne peut pas utiliser une anticipation littérale. Par exemple, dans un noeud Remplacer, vous ne pouvez pas remplacer la valeur de field1 par une expression comme @OFFSET(field1,-2).
@OFFSET(FIELD, EXPR, INT)	Tout	Effectue la même opération que la fonction @OFFSET, avec un troisième argument, <i>INT</i> , qui indique le nombre maximal de valeurs à prendre en compte. Si le décalage est calculé à partir d'une expression, ce troisième argument devrait améliorer les performances. Par exemple, dans une formule telle que @OFFSET(Foo, Month, 12), le système sait comment conserver uniquement les douze dernières valeurs de Foo. Sinon, il doit stocker chaque valeur à tout hasard. Pour les décalages dont la valeur est constante—y compris les décalages « anticipés » négatifs qui doivent être constants—le troisième argument n'est d'aucune utilité. Il est recommandé d'utiliser la version à deux arguments de cette fonction. Consultez également la note sur les fonctions autoréférentielles dans la version à deux arguments décrite plus tôt.
@SDEV(FIELD)	Réel	Renvoie l'écart-type des valeurs du ou des champs <i>FIELD</i> ou <i>FIELDS</i> spécifiés.
@SDEV(FIELD, EXPR)	Réel	Renvoie l'écart-type des valeurs du champ <i>FIELD</i> pour les <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, y compris l'enregistrement actuel. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, l'écart-type des enregistrements reçus est renvoyé.

Fonction	Résultat	Description
@SDEV(FIELD, EXPR, INT)	Réel	Renvoie l'écart-type des valeurs du champ <i>FIELD</i> pour les <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, y compris l'enregistrement actuel. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, l'écart-type des enregistrements reçus est renvoyé. <i>INT</i> indique le nombre maximal de valeurs à prendre en compte. Cette méthode est beaucoup plus efficace que si vous vous limitez à deux arguments.
@SINCE(EXPR)	Tout	Renvoie le nombre d'enregistrements qui ont été transmis depuis que l'expression CLEM arbitraire <i>EXPR</i> est devenue vraie.
@SINCE(EXPR, INT)	Tout	Lorsque vous ajoutez le second argument, <i>INT</i> , vous indiquez le nombre maximal d'enregistrements à prendre en compte. Si <i>EXPR</i> n'a jamais été vraie, <i>INT</i> est équivalent à @INDEX+1.
@SINCE0(EXPR)	Tout	Considère l'enregistrement en cours, contrairement à @SINCE. La fonction @SINCE0 renvoie la valeur 0 si l'expression <i>EXPR</i> est vraie pour l'enregistrement actuel.
@SINCE0(EXPR, INT)	Tout	Lorsque vous ajoutez le second argument, <i>INT</i> indique le nombre maximal d'enregistrements à prendre en compte.
@SUM(FIELD)	Nombre	Renvoie la somme des valeurs du ou des champs (<i>FIELD</i> ou <i>FIELDS</i>) spécifiés.
@SUM(FIELD, EXPR)	Nombre	Renvoie la somme des valeurs du champ <i>FIELD</i> pour les <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la somme des enregistrements reçus est renvoyée.
@SUM(FIELD, EXPR, INT)	Nombre	Renvoie la somme des valeurs du champ <i>FIELD</i> pour les <i>EXPR</i> (nombre représenté par une expression) derniers enregistrements reçus par le noeud actuel, enregistrement actuel compris. <i>FIELD</i> doit être le nom d'un champ numérique. <i>EXPR</i> peut être toute expression donnant comme résultat un entier supérieur à 0. Si l'expression <i>EXPR</i> est omise ou donne un résultat supérieur au nombre d'enregistrements reçus jusqu'à présent, la somme des enregistrements reçus est renvoyée. <i>INT</i> indique le nombre maximal de valeurs à prendre en compte. Cette méthode est beaucoup plus efficace que si vous vous limitez à deux arguments.
@THIS(FIELD)	Tout	Renvoie la valeur du champ <i>FIELD</i> dans l'enregistrement actuel. Utilisée uniquement dans les expressions @SINCE.

Fonctions globales

Les fonctions @MEAN, @SUM, @MIN, @MAX, et @SDEV fonctionnent avec, au plus, tous les enregistrements lus jusqu'à l'enregistrement en cours compris. Toutefois, dans certains cas, il est utile de comparer les valeurs de l'enregistrement actuel à celles de tout l'ensemble de données. Lorsque vous utilisez un noeud V. globales (Valeurs globales) pour générer des valeurs dans tout l'ensemble de données, vous pouvez accéder à ces valeurs dans une expression CLEM en utilisant les fonctions globales.

Par exemple,

@GLOBAL_MAX(Age)

renvoie la valeur maximale du champ Age dans l'ensemble de données, tandis que l'expression

$(\text{Value} - @GLOBAL_MEAN(\text{Value})) / @GLOBAL_SDEV(\text{Value})$

exprime la différence entre la valeur du champ Value dans l'enregistrement actuel et la moyenne globale, sous la forme d'un écart-type. Vous ne pouvez utiliser les valeurs globales qu'après leur calcul par un noeud V. globales (Valeurs globales). Vous pouvez annuler toutes les valeurs globales actuelles en cliquant sur le bouton Effacer les valeurs globales, dans l'onglet Valeurs globales de la boîte de dialogue des propriétés du flux.

Remarque : Il n'est pas possible d'appeler des fonctions @ à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
@GLOBAL_MAX(FIELD)	Nombre	Renvoie la valeur maximale du champ <i>FIELD</i> sur tout l'ensemble de données, tel qu'il a été précédemment généré par un noeud V. globales (Valeurs globales). <i>FIELD</i> doit être le nom d'un champ numérique. Une erreur se produit si la valeur globale correspondante n'a pas été définie. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
@GLOBAL_MIN(FIELD)	Nombre	Renvoie la valeur minimale du champ <i>FIELD</i> sur tout l'ensemble de données, tel qu'il a été précédemment généré par un noeud V. globales (Valeurs globales). <i>FIELD</i> doit être le nom d'un champ numérique. Une erreur se produit si la valeur globale correspondante n'a pas été définie.
@GLOBAL_SDEV(FIELD)	Nombre	Renvoie l'écart-type des valeurs du champ <i>FIELD</i> sur tout l'ensemble de données, tel qu'il a été précédemment généré par un noeud V. globales (Valeurs globales). <i>FIELD</i> doit être le nom d'un champ numérique. Une erreur se produit si la valeur globale correspondante n'a pas été définie.

Fonction	Résultat	Description
@GLOBAL_MEAN(FIELD)	Nombre	Renvoie la moyenne des valeurs du champ <i>FIELD</i> sur tout l'ensemble de données, tel qu'il a été précédemment généré par un noeud V. globales (Valeurs globales). <i>FIELD</i> doit être le nom d'un champ numérique. Une erreur se produit si la valeur globale correspondante n'a pas été définie.
@GLOBAL_SUM(FIELD)	Nombre	Renvoie la somme des valeurs du champ <i>FIELD</i> sur tout l'ensemble de données, tel qu'il a été précédemment généré par un noeud V. globales (Valeurs globales). <i>FIELD</i> doit être le nom d'un champ numérique. Une erreur se produit si la valeur globale correspondante n'a pas été définie.

Traitement des valeurs nulles et non renseignées

Le langage CLEM permet d'indiquer que certaines valeurs d'un champ doivent être considérées comme "blancs", ou manquantes. Vous pouvez utiliser les fonctions suivantes pour travailler avec des valeurs non renseignées.

Remarque : Il n'est pas possible d'appeler des fonctions @ à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
@BLANK(FIELD)	Booléen	Renvoie la valeur true (vrai) pour tous les enregistrements dont les valeurs sont vides, d'après les règles de traitement des blancs définies dans un noeud Typer ou un noeud source en amont (onglet Types). Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
@LAST_NON_BLANK(FIELD)	Tout	Renvoie la dernière valeur du champ <i>FIELD</i> qui était renseigné, tel que défini dans un noeud Typer ou source en amont. S'il n'existe aucune valeur renseignée pour le champ <i>FIELD</i> dans les enregistrements lus jusqu'à présent, la fonction renvoie \$null\$. Les valeurs non renseignées, également appelées valeurs manquantes utilisateur, peuvent être définies séparément pour chaque champ.
@NULL(FIELD)	Booléen	Renvoie la valeur true (vrai) si la valeur de <i>FIELD</i> est la valeur \$null\$, manquante dans le système. Renvoie la valeur false (faux) pour toutes les autres valeurs, y compris les valeurs non renseignées par les utilisateurs. Si vous souhaitez rechercher les deux valeurs, utilisez @BLANK(FIELD) et @NULL(FIELD).
undef	Tout	Cette fonction est généralement utilisée dans CLEM pour entrer une valeur \$null\$.—Par exemple, pour remplacer des valeurs non renseignées par des valeurs nulles dans le noeud Remplacer.

Les champs non renseignés peuvent être « remplis » avec le noeud Remplacer. Dans les noeuds Remplacer et Calculer (mode Multiple uniquement), la fonction spéciale CLEM@FIELD fait référence aux champs en cours.

Champs spéciaux

Les fonctions spéciales servent à signaler les champs spéciaux en cours d'analyse ou à générer une liste de champs en tant qu'entrée. Par exemple, lors du calcul simultané de plusieurs champs, utilisez @FIELD pour que le programme comprenne que « l'action de calcul doit porter sur les champs sélectionnés ». L'expression $\log(@FIELD)$ calcule un nouveau champ de logarithme pour chaque champ sélectionné.

Remarque : Il n'est pas possible d'appeler des fonctions @ à partir de scripts. [Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.](#)

Fonction	Résultat	Description
@FIELD	Tout	Effectue une action sur tous les champs spécifiés dans le contexte de l'expression. Il est impossible d'appeler cette fonction à partir d'un script. Pour plus d'informations, reportez-vous à la section Expressions CLEM dans les scripts dans le chapitre 3 sur p. 29.
@TARGET	Tout	Quand une expression CLEM est utilisée dans une fonction d'analyse définie par l'utilisateur, @TARGET représente le champ cible ou la « valeur correcte » de la paire cible/prédite analysée. Cette fonction est fréquemment utilisée dans les noeuds Analyse.
@PREDICTED	Tout	Quand une expression CLEM est utilisée dans une fonction d'analyse définie par l'utilisateur, @PREDICTED représente la valeur prédite de la paire cible/prédite analysée. Cette fonction est fréquemment utilisée dans les noeuds Analyse.
@PARTITION_FIELD	Tout	Remplace le nom du champ de partition actuel.
@TRAINING_PARTITION	Tout	Renvoie la valeur de la partition d'apprentissage actuelle. Par exemple, pour sélectionner des noeuds d'apprentissage utilisant un noeud Sélectionner, utilisez l'expression CLEM suivante : @PARTITION_FIELD = @TRAINING_PARTITION Ainsi, le noeud Sélectionner sera toujours fonctionnel, quelles que soient les valeurs utilisées pour représenter chaque partition des données.
@TESTING_PARTITION	Tout	Renvoie la valeur de la partition de test actuelle.
@VALIDATION_PARTITION	Tout	Renvoie la valeur de la partition de validation actuelle.

Fonction	Résultat	Description
@FIELDS_BETWEEN(start, end)	<i>Tout</i>	Renvoie la liste de noms des champs situés entre les champs de début et de fin spécifiés (ces champs étant inclus), en fonction de l'ordre naturel (c'est-à-dire l'ordre d'insertion) des champs dans les données. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans <i>Guide de l'utilisateur de IBM SPSS Modeler 15</i>.
@FIELDS_MATCHING(pattern)	<i>Tout</i>	Renvoie la liste des noms de champ qui correspondent à un patron spécifié. Il est possible d'inclure un point d'interrogation (?) dans le patron pour obtenir une correspondance exacte avec un caractère ; un astérisque (*) implique une correspondance avec zéro ou plusieurs caractères. Pour obtenir une correspondance avec un point d'interrogation ou un astérisque de littéral (plutôt que de les utiliser comme caractères génériques), il est possible d'utiliser une barre oblique inverse (\) en tant que caractère d'échappement. Pour plus d'informations, reportez-vous à la section Récapitulation de plusieurs champs dans le chapitre 7 dans <i>Guide de l'utilisateur de IBM SPSS Modeler 15</i>.
@MULTI_RESPONSE_SET	<i>Tout</i>	Renvoie la liste des champs dans l'ensemble à réponses multiples nommé. Pour plus d'informations, reportez-vous à la section Utilisation des données à réponses multiples dans le chapitre 7 dans <i>Guide de l'utilisateur de IBM SPSS Modeler 15</i>.

Partie II:
Référence sur les propriétés

Référence sur les propriétés

Introduction aux références sur les propriétés

Vous pouvez indiquer un certain nombre de propriétés pour les noeuds, les flux, les super noeuds et les projets. Certaines propriétés sont communes à tous les noeuds, telles que le nom, l'annotation et l'info-bulle, alors que d'autres sont propres à certains types de noeud. D'autres propriétés font référence aux opérations de flux de haut niveau, comme la mise en cache ou le comportement du super noeud. Vous pouvez accéder aux propriétés via l'interface utilisateur standard (par exemple, lorsque vous ouvrez une boîte de dialogue pour modifier les options d'un noeud) et les utiliser de plusieurs manières.

- Vous pouvez modifier les propriétés via des scripts, comme l'explique cette section. Pour plus d'informations, reportez-vous à Syntaxe des propriétés ci-dessous.
- Les propriétés de noeud peuvent être utilisées dans les paramètres du super noeud. [Pour plus d'informations, reportez-vous à la section Utilisation des paramètres de super noeud pour accéder aux propriétés du noeud dans le chapitre 9 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)
- Les propriétés de noeud peuvent également être utilisées dans le cadre d'une option de ligne de commande (en utilisant le commutateur -P) lors du démarrage de IBM® SPSS® Modeler.

Dans le cadre de la génération de scripts dans SPSS Modeler, les propriétés de noeud et de flux sont souvent appelées **paramètres de propriété**. Dans cette aide, elles sont appelées propriétés de noeud ou de flux.

Pour plus d'informations sur le langage de script, reportez-vous au le chapitre 3.

Syntaxe des propriétés

Les propriétés doivent utiliser la structure syntaxique suivante :

```
NAME:TYPE.PROPERTY
```

où **NAME** est le nom d'un noeud et **TYPE** est son type (par exemple, `multiplotnode` ou `derivnode`). Vous pouvez ignorer **NAME** ou **TYPE**, mais vous devez inclure au moins l'un d'eux. **PROPERTY** est le nom du noeud ou du paramètre de flux auquel votre expression fait référence. Par exemple, la syntaxe suivante est utilisée pour filtrer le champ *Age* à partir des données en aval :

```
set mynode:filternode.include.Age = false
```

Pour utiliser une valeur personnalisée pour un des paramètres (**NAME**, **TYPE**, ou **PROPERTY**), définissez d'abord la valeur dans une instruction, comme `set derive.new_name = mynewfield`. Vous pouvez ensuite utiliser la valeur `mynewfield` comme paramètre en la faisant précéder du

symbole `^`. Par exemple, vous pouvez définir le type du noeud Calculer nommé ci-avant, en utilisant la syntaxe suivante :

```
set ^mynewfield.result_type = "Conditional"
```

Tous les noeuds utilisés dans IBM® SPSS® Modeler peuvent être spécifiés dans le paramètre TYPE de la syntaxe NAME:TYPE.PROPERTY.

Propriétés structurées

Pour obtenir une plus grande clarté lors de l'analyse, la génération de scripts peut utiliser les propriétés structurées de deux façons différentes :

- Attribuer une structure aux noms des propriétés des noeuds complexes, tels que les noeuds Typer, Filtrer ou Equilibrer.
- Fournir un format afin de spécifier plusieurs propriétés en une opération.

Structuration des interfaces complexes

Les scripts des noeuds contenant des tableaux ou d'autres interfaces complexes (par exemple, les noeuds Typer, Filtrer et Equilibrer) doivent suivre une structure particulière afin de procéder à une analyse correcte. Le nom des propriétés structurées reflète cette complexité. Par exemple, dans un noeud Filtrer, chaque champ disponible (en amont) peut être activé ou désactivé. Pour faire référence à ces informations, le noeud Filtrer stocke une information par champ (indiquant s'il est vrai ou faux). Ces champs multiples sont accessibles et modifiables via une propriété unique appelée **field** (champ). Cette propriété peut avoir, ou se voir affecter, la valeur *true* ou *false*. Prenons l'exemple d'un noeud Filtrer appelé *mynode* et comportant, en amont, un champ appelé *Age*. Pour désactiver ce champ, définissez la propriété *mynode.include.Age* sur la valeur *false* (faux), de la façon suivante :

```
set mynode.include.Age = false
```

Structuration permettant de définir plusieurs propriétés

Pour de nombreux noeuds, vous pouvez affecter plusieurs propriétés de noeud ou de flux à la fois. Cette opération est appelée **définition globale** ou **définition en bloc**. [Pour plus d'informations, reportez-vous à la section Commande set dans le chapitre 4 sur p. 34.](#)

Les propriétés structurées peuvent parfois être très complexes. La barre oblique inverse (`\`) peut être utilisée comme caractère de continuation de ligne pour vous aider à aligner les arguments avec plus de clarté. Voir l'exemple comme suit :

```
mynode:sortnode.keys = [{ 'K' Descending} \
                        { 'Age' Ascending}\
                        { 'Na' Descending }]
```

Les propriétés structurées offrent également la possibilité de définir plusieurs propriétés d'un noeud avant que le noeud soit stable. Par défaut, une définition globale définit toutes les propriétés d'un bloc avant d'agir en fonction du paramétrage d'une propriété spécifique. Par exemple, lors de la définition d'un noeud Fixe, la spécification des propriétés de champs en deux étapes

génère des erreurs dans la mesure où le noeud n'est pas cohérent tant que les deux paramètres ne sont pas valides. Le fait de définir les propriétés en utilisant une définition globale permet d'éviter ce problème dans la mesure où les deux propriétés sont définies avant que le modèle de données ne soit mis à jour.

Abréviations

Les abréviations standard sont utilisées tout au long de la syntaxe des propriétés de noeud. La connaissance des abréviations est utile lors de la rédaction de scripts.

Abréviation	Signification
abs	Absolute value
len	Longueur
min	Minimum
max	Maximum
correl	Correlation
covar	Covariance :
num	Nombre ou numérique
pct (pourcentage)	Percent (pour cent) ou (percentage) pourcentage
transp	Transparence
xval	Cross-validation (validation croisée)
var	Variance ou variable (dans les noeuds source)

Exemple de propriétés de noeud et de flux

Dans IBM® SPSS® Modeler, les propriétés de noeud et de flux peuvent être utilisées de différentes façons. Ces propriétés sont le plus souvent utilisées dans le cadre d'un script, que ce soit un **script autonome**, pour automatiser plusieurs flux ou opérations, ou un **script de flux**, pour automatiser des processus au sein d'un flux unique. Vous pouvez également indiquer des paramètres de noeud à l'aide des propriétés de noeud du super noeud. Au niveau le plus basique, les propriétés peuvent aussi être utilisées sous forme d'une option de ligne de commande pour démarrer SPSS Modeler. Si vous utilisez le commutateur -p dans le cadre de l'appel d'une ligne de commande, vous pouvez modifier un paramètre dans le flux à l'aide d'une propriété du flux.

s.max_size	Désigne la propriété max_size du noeud s.
s:samplenode.max_size	Désigne la propriété max_size du noeud s, qui doit être un noeud Echantillonner.
:samplenode.max_size	Désigne la propriété max_size du noeud Echantillonner du flux actuel (il ne peut exister qu'un noeud de ce type).
s:sample.max_size	Désigne la propriété max_size du noeud s, qui doit être un noeud Echantillonner.
t.direction.Age	Désigne le rôle du champ Age du noeud Typier t.
:.max_size	*** NON AUTORISE *** Vous devez indiquer le nom ou le type du noeud.

Comme l'illustre l'exemple `s:sample.max_size`, vous n'avez pas besoin d'écrire les types de noeud en entier.

L'exemple `t.direction.Age` montre que certains noms de propriété peuvent être structurés lorsque les attributs d'un noeud sont trop complexes pour pouvoir être définis par des propriétés aux valeurs individuelles. Ces propriétés sont appelées propriétés **structurées** ou **complexes**.

Présentation des propriétés de noeud

Pour chaque type de noeud, un certain nombre de propriétés sont autorisées. Chaque propriété comporte un type. Ce type peut être général—nombre, booléen ou chaîne—auquel cas le type approprié est attribué automatiquement aux paramètres de la propriété. Si le type correct ne peut pas être défini, une erreur est renvoyée. Un paramètre de propriété peut également spécifier un intervalle de valeurs autorisées, telles que `Discard`, `PairAndDiscard` et `IncludeAsText`. Dans ce cas, une erreur est renvoyée si une autre valeur est utilisée. Les propriétés booléennes doivent être lues ou définies avec les valeurs `true` et `false` (vrai et faux). (Les variations qui contiennent `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `false`, `False`, `FALSE`, ou `0` sont également reconnues lors de la définition des valeurs mais peuvent parfois générer des erreurs lors de la lecture des valeurs de propriété. Toute autre valeur sera interprétée comme étant `True` (vrai). Utiliser `true` et `false` de façon cohérente permet d'éviter toute confusion.) Dans les tableaux de référence du présent document, les propriétés structurées sont indiquées telles quelles dans la colonne *Description de la propriété* et sont accompagnées de la syntaxe à utiliser.

Propriétés communes des noeuds

Dans IBM® SPSS® Modeler, un certain nombre de propriétés sont communes à tous les noeuds (y compris les super noeuds).

Nom de la propriété	Le type de données	Description de la propriété
<code>use_custom_name</code>	<i>Booléen</i>	
<code>name</code>	<i>chaîne</i>	Propriété en lecture seule qui lit le nom (automatique ou personnalisé) d'un noeud de l'espace de travail.
<code>custom_name</code>	<i>chaîne</i>	Indique le nom personnalisé du noeud.
<code>tooltip</code>	<i>chaîne</i>	
<code>annotation</code>	<i>chaîne</i>	
<code>keywords</code>	<i>chaîne</i>	Propriété structurée indiquant la liste de mots-clés associés à l'objet (par exemple, ["Keyword1" "Keyword2"]).
<code>cache_enabled</code>	<i>Booléen</i>	
<code>node_type</code>	<code>source_supernode</code> <code>process_supernode</code> <code>terminal_supernode</code> tous les noms de noeud indiqués pour la génération de scripts	Propriété en lecture seule utilisée pour faire référence à un noeud par type. Par exemple, au lieu de faire référence à un noeud uniquement par son nom, tel que <code>real_income</code> , vous pouvez également indiquer son type, tel que (<code>userinputnode</code> ou <code>filternode</code>).

Les propriétés propres aux super noeuds sont abordées séparément, comme pour tous les autres noeuds. [Pour plus d'informations, reportez-vous à la section Propriétés du super noeud dans le chapitre 22 sur p. 320.](#)

Propriétés du flux

La génération de scripts permet de contrôler différentes propriétés de flux. Pour référencer les propriétés de flux, vous devez utiliser une variable de flux spéciale et placer le symbole ^ devant le flux :

```
set ^stream.execute_method = Script
```

Exemple

La propriété nodes est utilisée pour faire référence aux noeuds du flux en cours. Le script de flux suivant fournit un exemple :

```
var listofnodes
var thenode
set listofnodes = ^stream.nodes

set ^stream.annotation = ^stream.annotation >< "\n\nCe flux est appelé \"\" >< ^stream.name >
< \"\" and contains/ the following nodes\n"

for thenode in listofnodes
set ^stream.annotation = ^stream.annotation >< "\n" >< ^thenode.node_type
endfor
```

Dans l'exemple ci-dessus, la propriété nodes est utilisée pour créer la liste de tous les noeuds du flux et pour écrire cette liste dans les annotations du flux. L'annotation générée ressemble à l'annotation suivante :

Ce flux est appelé « druglearn » et contient les noeuds suivants

```
derivenode
neuralnetworknode
variablefilenode
typenode
c50node
filternode
```

Les propriétés de flux sont décrites dans le tableau suivant.

Nom de la propriété	Le type de données	Description de la propriété
execute_method	Normal Script	
date_format	"JJMMAA" "MMJJAA" "AAMMJJ" "AAAAMMJJ" "AAAAJJJ" DAY MONTH "JJ-MM-AA" "JJ-MM-AAAA" "MM-JJ-AA" "MM-JJ-AAAA" "JJ-MOI-AA" "JJ-MOI-AAAA" "AAAA-MM-JJ" "JJ.MM.AA" "JJ.MM.AAAA" "MM.JJ.AA" "MM.JJ.AAAA" "JJ.MOI.AA" "JJ.MOI.AAAA" "JJ/MM/AA" "JJ/MM/AAAA" "MM/JJ/AA" "MM/JJ/AAAA" "JJ/MOI/AA" "JJ/MOI/AAAA" MOIS AAAA t T AAAA ss SM AAAA	
date_baseline	<i>nombre</i>	
date_2digit_baseline	<i>nombre</i>	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
time_rollover	<i>booléen</i>	
import_datetime_as_string	<i>booléen</i>	
decimal_places	<i>nombre</i>	

Nom de la propriété	Le type de données	Description de la propriété
decimal_symbol	Default Period Comma	
angles_in_radians	<i>booléen</i>	
use_max_set_size	<i>booléen</i>	
max_set_size	<i>nombre</i>	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	<i>booléen</i>	Permet de rafraîchir les noeuds source automatiquement dès l'exécution du flux.
script	<i>chaîne</i>	
annotation	<i>chaîne</i>	Exemple : set ^stream.annotation = "something interesting"
name	<i>chaîne</i>	Exemple : set x = ^stream.name <i>Remarque</i> : Cette propriété est en lecture seule. Si vous souhaitez modifier le nom d'un flux, vous devez l'enregistrer sous un autre nom.
parameters		Utilisez cette propriété pour mettre à jour les paramètres de flux à partir d'un script autonome. Exemple : set ^stream.parameters.height = 23
nodes		Reportez-vous aux informations détaillées ci-dessous.
encoding	SystemDefault "UTF-8"	
stream_rewriting	<i>booléen</i>	
stream_rewriting_maximise_sql	<i>booléen</i>	
stream_rewriting_optimise_clem_execution	<i>booléen</i>	
stream_rewriting_optimise_syntax_execution	<i>booléen</i>	
enable_parallelism	<i>booléen</i>	
sql_generation	<i>booléen</i>	
database_caching	<i>booléen</i>	
sql_logging	<i>booléen</i>	
sql_generation_logging	<i>booléen</i>	
sql_log_native	<i>booléen</i>	
sql_log_prettyprint	<i>booléen</i>	
record_count_suppress_input	<i>booléen</i>	
record_count_feedback_interval	<i>entier</i>	

Propriétés du projet

Un certain nombre de propriétés sont disponibles avec les projets pour la génération de scripts.

Exemple

```
load project "C:/clemdata/DrugData.cpj"
set ^project.summary="Initial modeling work on the latest drug data."
set ^project.ordering=NameAddedType
execute _project
```

Nom de la propriété	Le type de données	Description de la propriété
summary	<i>chaîne</i>	Récapitulatif du projet (généralement une version abrégée de l'annotation).
title	<i>chaîne</i>	Titre du rapport.
author	<i>chaîne</i>	Auteur du rapport.
structure	Phase Class	Détermine le mode d'organisation du projet—par phase de Data mining ou par type d'objet (classe).
include_mode	IncludedItems ExcludedItems AllItems	Détermine les éléments à inclure dans le rapport du projet.
select_mode	AllItems RecentItems OldItems	Détermine (en fonction de l'ancienneté) les éléments à inclure dans le rapport.
recent_item_limit	<i>entier</i>	Utilisé lorsque <code>select_mode</code> est <code>RecentItems</code>
old_item_limit	<i>entier</i>	Utilisé lorsque <code>select_mode</code> est <code>OldItems</code>
ordering	TypeNameAdded TypeAddedName NameAddedType AddedNameType	Détermine l'ordre dans lequel les éléments apparaissent dans le rapport.

Propriétés des nœuds source

Propriétés communes aux nœuds source

Les propriétés communes à tous les nœuds source sont répertoriées ci-dessous, avec des informations sur certains nœuds dans les rubriques suivantes.

Exemple

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.use_custom_values.Age = True
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
#storage is read only
set :variablefilenode.check.Age = None
set :variablefilenode.values.Age = [1 100]
```

Nom de la propriété	Le type de données	Description de la propriété
direction	Input Target Both None Partition Split Frequency RecordID	Propriété saisie (rôles des champs). Syntaxe : NODE.direction.FIELDNAME <i>Remarque</i> : Les valeurs In et Out sont désormais remplacées. Leur prise en charge pourrait être supprimée dans une version ultérieure.
type	Range Flag Set Typeless Discrete Ordered Set Default	Type de champ. Si vous paramétrez cette propriété sur <i>Par défaut</i> , toute définition de propriété de <i>values</i> est effacée. Si la valeur <i>value_mode</i> est définie sur <i>Spécifier</i> , elle est redéfinie sur <i>Lire</i> . Si la propriété <i>value_mode</i> est paramétrée sur <i>Transférer</i> ou <i>Lire</i> , la définition du <i>type</i> n'a aucune incidence sur elle. Syntaxe : NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	Propriété saisie en lecture seule pour le type de stockage de champ. Syntaxe : NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	Propriété saisie (vérification du type et de l'intervalle des champs). Syntaxe : NODE.check.FIELDNAME

Nom de la propriété	Le type de données	Description de la propriété
values	[<i>valeur valeur</i>]	Pour un champ continu (intervalle), la première valeur représente la valeur minimale et la dernière valeur, la valeur maximale. Pour des champs nominaux (ensemble), spécifiez toutes les valeurs. Pour les champs booléens, la première valeur représente <i>false</i> (faux) et la dernière valeur, <i>true</i> (vrai). La définition automatique de cette propriété paramètre la propriété <i>value_mode</i> sur <i>Spécifier</i> . Syntaxe : NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Détermine la façon dont les valeurs sont déterminées pour un champ lors du passage suivant des données. Syntaxe : NODE.value_mode.FIELDNAME Remarque : vous ne pouvez pas paramétrer directement cette propriété sur <i>Spécifier</i> . Pour utiliser des valeurs spécifiques, paramétrez la propriété <i>values</i> .
default_value_mode	Read Pass	Indique la méthode par défaut de définition des valeurs de tous les champs. Syntaxe : NODE.default_value_mode Exemple : set mynode.default_value_mode = Pass Ce paramètre peut être ignoré pour certains champs à l'aide de la propriété <i>value_mode</i> .
extend_values	<i>booléen</i>	S'applique quand <i>value_mode</i> est défini sur <i>Lire</i> . Paramétrez cette valeur sur <i>T</i> pour ajouter des valeurs qui viennent d'être lues aux valeurs existantes du champ. Paramétrez cette valeur sur <i>F</i> pour supprimer des valeurs existantes en faveur des valeurs qui viennent d'être lues. Syntaxe : NODE.extend_values.FIELDNAME
value_labels	<i>chaîne</i>	Permet d'utiliser une étiquette de valeur. Exemple : set :varfilenode.value_labels.Age = [{3 three}{5 five}] Vous devez d'abord indiquer les valeurs.
enable_missing	<i>booléen</i>	Lorsque cette propriété est paramétrée sur <i>T</i> , elle active le suivi des valeurs manquantes du champ. Syntaxe : NODE.enable_missing.FIELDNAME
missing_values	[<i>valeur valeur ...</i>]	Spécifie les valeurs de données qui indiquent les données manquantes. Syntaxe : NODE.missing_values.FIELDNAME
range_missing	<i>booléen</i>	Lorsque cette propriété est définie sur <i>T</i> , spécifie si un intervalle de valeurs manquantes (vides) est défini pour un champ. Syntaxe : NODE.range_missing.FIELDNAME
missing_lower	<i>chaîne</i>	Lorsque le paramètre <i>range_missing</i> a pour valeur <i>true</i> (vrai), indique la limite inférieure de l'intervalle des valeurs manquantes. Syntaxe : NODE.missing_lower.FIELDNAME

Nom de la propriété	Le type de données	Description de la propriété
missing_upper	chaîne	Lorsque le paramètre <code>range_missing</code> a pour valeur <code>true</code> (vrai), indique la limite supérieure de l'intervalle des valeurs manquantes. Syntaxe : <code>NODE.missing_upper.FIELDNAME</code>
null_missing	booléen	Lorsque cette propriété est paramétrée sur <i>T</i> , les valeurs nulles (valeurs non définies affichées sous la forme <code>\$null\$</code> dans le logiciel) sont considérées comme des valeurs manquantes. Syntaxe : <code>NODE.null_missing.FIELDNAME</code>
whitespace_missing	booléen	Lorsque cette propriété est paramétrée sur <i>T</i> , les valeurs composées uniquement d'espaces blancs (espaces, tabulations et caractères de nouvelle ligne) sont considérées comme des valeurs manquantes. Syntaxe : <code>NODE.whitespace_missing.FIELDNAME</code>
description	chaîne	Permet d'indiquer une étiquette de champ ou une description.
default_include	booléen	Propriété saisie permettant d'indiquer si le comportement par défaut consiste à transmettre ou à filtrer les champs : <code>NODE.default_include</code> Exemple : <code>set mynode:filternode.default_include = false</code>
include	booléen	Propriété saisie utilisée pour déterminer pour chaque champ s'il doit être inclus ou filtré : <code>NODE.include.FIELDNAME</code> Exemple : <code>set mynode:filternode.include.Age = true</code>
new_name	chaîne	Exemple : <code>set mynode:filternode.new_name.'Age' = "years"</code>

propriétés de *cognosimportnode*



Le noeud source IBM Cognos BI importe des données depuis les bases de données Cognos BI. [Pour plus d'informations, reportez-vous à la section Importation des données Cognos dans le chapitre 2 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create cognosimportnode
set :cognosimportnode.cognos_connection = {'http://mycogsrv1:9300/p2pd/servlet/dispatch', true, "", "", ""}
set :cognosimportnode.cognos_package_name = '/Public Folders/GOSALES'
set :cognosimportnode.cognos_items = {"[GreatOutdoors].[BRANCH].[BRANCH_CODE]"}
```

"[GreatOutdoors].[BRANCH].[COUNTRY_CODE]"]

propriétés cognosimportnode	Le type de données	Description de la propriété
mode	Data Report	Spécifie s'il faut importer les données Cognos BI (par défaut) ou les rapports.
cognos_connection	{ <i>"field"</i> , <i>"field"</i> , ... <i>"field"</i> }	Une propriété de liste contenant les détails de la connexion du serveur Cognos. Le format est le suivant : { <i>"URL_Serveur_Cognos"</i> , <i>Mode_connexion</i> , <i>"espace de nommage"</i> , <i>"nom utilisateur"</i> , <i>"mot de passe"</i> } où : <i>URL_Serveur_Cognos</i> est l'URL du serveur Cognos vers lequel vous effectuez l'exportation <i>mode_connexion</i> indique si la connexion anonyme est utilisée et est soit true soit false; si les champs suivants sont définis sur true, ils doivent être définis sur "" <i>espace de nommage</i> spécifie le fournisseur de sécurité pour l'authentification utilisé pour se connecter au serveur. <i>nom utilisateur</i> et <i>mot de passe</i> sont ceux utilisés pour la connexion au serveur Cognos
cognos_package_name	chaîne	Le chemin d'accès et le nom du package Cognos depuis lequel vous importez les objets de données, par exemple : /Public Folders/GOSALES Remarque : seules les barres obliques sont valides.
cognos_items	{ <i>"field"</i> , <i>"field"</i> , ... <i>"field"</i> }	Le nom d'un ou plusieurs objets de données à importer. Le format de <i>champ</i> est [<i>namespace</i>].[<i>query_subject</i>].[<i>query_item</i>] Exemple : set :cognosimport.cognos_items = {"[Inventory (query)].[Inventory].[Opening inventory]", "[Inventory (query)].[Inventory].[Quantity shipped]", "[Inventory (query)].[Inventory].[Additions]", "[Inventory (query)].[Inventory].[Unit cost]", "[Inventory (query)].[Inventory].[Closing inventory]", "[Inventory (query)].[Inventory].[Average unit cost]"}
cognos_filters	champ	Le nom d'un ou plusieurs filtres à appliquer avant d'importer les données. Exemple : set :cognosimport.cognos_filters = {"[Inventory].[Filter].[MyFilter]"}
cognos_data_parameters	Liste	Valeurs des paramètres d'invite des données. Les paires nom/valeur sont entre accolades et les paires multiples sont séparées par des virgules et la chaîne entière est entre crochets. Format : [{" <i>param1</i> ", " <i>value</i> "}, ..., {" <i>paramN</i> ", " <i>value</i> "}] Exemple : set :cognosimport.cognos_data_parameters = [{"SexValue", "F"}, {"a", "1"}, {"b", "1"}]

propriétés cognosimportnode	Le type de données	Description de la propriété
cognos_report_location	champ	Le chemin Cognos d'un dossier ou d'un package depuis lequel importer les rapports, par exemple : /Public Folders/GOSALES Remarque : seules les barres obliques sont valides.
cognos_report_name	champ	Le chemin et le nom dans l'emplacement d'un rapport à importer par exemple : set :cognosimport.cognos_report_name = /Jimmy/Package/Drug4nPackage/3columns
cognos_report_parameters	Liste	Valeurs des paramètres de rapport. Les paires nom/valeur sont entre accolades et les paires multiples sont séparées par des virgules et la chaîne entière est entre crochets. Format : [{"param1", "value"}, ..., {"paramN", "value"}] Exemple : set :cognosimport.cognos_report_parameters = [{"SexValue", "F"}, {"a", "1"}, {"b", "1"}]

Propriétés de databasenode



Le nœud SGBD peut être utilisé pour importer des données provenant de nombreux autres logiciels utilisant la connectivité ODBC (Open Database Connectivity), tels que Microsoft SQL Server, DB2, Oracle, etc. [Pour plus d'informations, reportez-vous à la section Nœud Source de base de données dans le chapitre 2 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create databasenode
set :databasenode.mode = Table
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = "Drug4n_db"
set :databasenode.username = "spss"
set :databasenode.password = "spss"
var test_e
set test_e = :databasenode.epassword
set :databasenode.tablename = ".Drug4n"
```

propriétés databasenode	Le type de données	Description de la propriété
mode	Table Query	Spécifiez <i>Table</i> pour établir la connexion à une table de base de données à l'aide des commandes de la boîte de dialogue ou indiquez <i>Requête</i> pour interroger la base de données sélectionnée en utilisant SQL.
datasource	chaîne	Nom de la base de données (reportez-vous à la remarque ci-dessous).

propriétés databasenode	Le type de données	Description de la propriété
username	chaîne	Détails de la connexion à la base de données (reportez-vous aussi à la remarque ci-dessous).
password	chaîne	
epassword	chaîne	Indique un mot de passe codé à la place du codage en dur d'un mot de passe dans un script. Pour plus d'informations, reportez-vous à la section Génération d'un mot de passe codé dans le chapitre 5 sur p. 62. Cette propriété est en lecture seule au cours de l'exécution.
tablename	chaîne	Nom de la table à laquelle vous souhaitez accéder.
strip_spaces	None Left Right Both	Options permettant de supprimer des espaces situés en début et en fin de chaîne.
use_quotes	AsNeeded Always Never	Indiquez si les noms des tables et des colonnes sont placés entre guillemets lors de l'envoi des requêtes à la base de données (par exemple, s'ils contiennent des espaces ou des signes de ponctuation).
query	chaîne	Indique le code SQL de la requête à soumettre.

Remarque : Si le nom de la base de données (dans la propriété de datasource) comporte des espaces, au lieu des propriétés individuelles de datasource, username et password, il est alors préférable d'utiliser une propriété de datasource unique au format suivant :

propriétés databasenode	Le type de données	Description de la propriété
datasource	chaîne	Format : {database_name,username,password[,true false]} Le dernier paramètre sert aux mots de passe codés. S'il est défini sur true, le mot de passe est décodé avant usage.

Exemple

```
create databasenode
set :databasenode.mode = Table
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = {"ORA 10gR2", user1, mypsw, true}
var test_e
set test_e = :databasenode.epassword
set :databasenode.tablename = ".Drug4n"
```

Utilisez aussi ce format si vous modifiez la source de données ; cependant, si vous souhaitez simplement changer le nom d'utilisateur ou le mot de passe, vous pouvez utiliser les propriétés username ou password.

Propriétés de `datacollectionimportnode`



Le nœud d'importation des données IBM® SPSS® Data Collection importe des données d'enquête en fonction du modèle Data Collection Data Model utilisé par les produits IBM Corp. dédiés aux études de marché. Pour pouvoir utiliser ce nœud, vous devez avoir installé avant la bibliothèque de données Data Collection. [Pour plus d'informations, reportez-vous à la section Noeud Data Collection dans le chapitre 2 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create datacollectionimportnode
set :datacollectionimportnode.metadata_name="mrQvDsc"
set :datacollectionimportnode.metadata_file="C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.casedata_name="mrQvDsc"
set :datacollectionimportnode.casedata_source_type=File
set :datacollectionimportnode.casedata_file="C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.import_system_variables = Common
set :datacollectionimportnode.import_multi_response = MultipleFlags
```

propriétés <code>datacollectionimportnode</code>	Le type de données	Description de la propriété
<code>metadata_name</code>	<i>chaîne</i>	Nom du MDSC. La valeur spéciale <code>DimensionsMDD</code> indique que le document de métadonnées Data Collection standard doit être utilisé. Autres valeurs possibles : <code>mrADODsc</code> <code>mrI2dDsc</code> <code>mrLogDsc</code> <code>mrQdiDrDsc</code> <code>mrQvDsc</code> <code>mrSampleReportingMDSC</code> <code>mrSavDsc</code> <code>mrSCDsc</code> <code>mrScriptMDSC</code> La valeur spéciale <code>none</code> (aucun) indique qu'il n'existe aucun MDSC.
<code>metadata_file</code>	<i>chaîne</i>	Nom du fichier de stockage des métadonnées.
<code>casedata_name</code>	<i>chaîne</i>	Nom du CDSC. Valeurs possibles : <code>mrADODsc</code> <code>mrI2dDsc</code> <code>mrLogDsc</code> <code>mrPunchDSC</code> <code>mrQdiDrDsc</code> <code>mrQvDsc</code> <code>mrRdbDsc2</code> <code>mrSavDsc</code> <code>mrScDSC</code> <code>mrXmlDsc</code> La valeur spéciale <code>none</code> (aucun) indique qu'il n'existe aucun CDSC.

propriétés datacollectionimportnode	Le type de données	Description de la propriété
casedata_source_type	Unknown File Folder UDL DSN	Indique le type de source du CDSC.
casedata_file	chaîne	Quand la propriété casedata_source_type est paramétrée sur <i>Fichier</i> , cette propriété indique le fichier contenant les données d'observation.
casedata_folder	chaîne	Quand la propriété casedata_source_type est paramétrée sur <i>Dossier</i> , cette propriété indique le dossier contenant les données d'observation.
casedata_udl_string	chaîne	Quand la propriété casedata_source_type est paramétrée sur <i>UDL</i> , cette propriété indique la chaîne de connexion OLE-DB correspondant à la source de données contenant les données d'observation.
casedata_dsn_string	chaîne	Quand la propriété casedata_source_type est paramétrée sur <i>DSN</i> , cette propriété indique la chaîne de connexion ODBC correspondant à la source de données.
casedata_project	chaîne	Lorsque vous lisez des données d'observation provenant d'une base de données Data Collection, vous pouvez fournir le nom du projet. Pour tous les autres types de données d'observation, ce paramètre doit rester vide.
version_import_mode	All Latest Specify	Définit le mode de traitement des versions.
specific_version	chaîne	Quand la propriété version_import_mode est paramétrée sur <i>Spécifier</i> , cette propriété définit la version des données d'observation à importer.
use_language	chaîne	Définit si les étiquettes d'une langue spécifique doivent être utilisées.
language	chaîne	Si la propriété use_language est paramétrée sur True (vrai), cette propriété définit le code langue à utiliser pour l'importation. Il doit s'agir de l'un des codes disponibles dans les données d'observation.
use_context	chaîne	Définit si un contexte spécifique doit être importé. Les contextes sont utilisés pour varier la description associée aux réponses.
context	chaîne	Si la propriété use_context est paramétrée sur True (vrai), cette propriété définit le contexte à importer. Il doit s'agir de l'un des contextes disponibles dans les données d'observation.
use_label_type	chaîne	Définit si un type d'étiquette spécifique doit être importé.

propriétés datacollectionimportnode	Le type de données	Description de la propriété
label_type	chaîne	Si la propriété use_label_type est paramétrée sur True (vrai), cette propriété définit le type d'étiquette à importer. Il doit s'agir de l'un des types d'étiquette disponibles dans les données d'observation.
user_id	chaîne	Pour les bases de données exigeant une connexion explicite, indiquez l'ID et le mot de passe utilisateur nécessaires pour accéder à la source de données.
password	chaîne	
import_system_variables	Common None All	Spécifie les variables système qui sont importées.
import_codes_variables	booléen	
import_sourcefile_variables	booléen	
import_multi_response	MultipleFlags Single	

Propriétés de excelimportnode



Le nœud Import Excel permet d'importer des données issues de n'importe quelle version de Microsoft Excel. Aucune source de données ODBC n'est requise. [Pour plus d'informations, reportez-vous à la section Noeud source Excel dans le chapitre 2 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
#Pour utiliser un intervalle nommé :
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.use_named_range = True
set :excelimportnode.named_range = "DRUG"
set :excelimportnode.read_field_names = True
```

```
#Pour utiliser un intervalle explicite :
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.worksheet_mode = Name
set :excelimportnode.worksheet_name = "Drug"
set :excelimportnode.explicit_range_start = A1
```

set :excelimportnode.explicit_range_end = F300

propriétés excelimportnode	Le type de données	Description de la propriété
excel_file_type	Excel2003 Excel2007	
full_filename	chaîne	Nom du fichier complet, y compris son chemin d'accès.
use_named_range	Booléen	Si vous voulez utiliser un intervalle nommé. Si cette propriété est définie sur True (vrai), la propriété named_range est utilisée pour indiquer l'intervalle à lire. Les autres paramètres de feuille de calcul et d'intervalle de données sont ignorés.
named_range	chaîne	
worksheet_mode	Index Name	Indique si la feuille de calcul est définie par un index ou par un nom.
worksheet_index	entier	Index de la feuille de calcul à lire : 0 pour la première, 1 pour la deuxième, etc.
worksheet_name	chaîne	Nom de la feuille de calcul à lire.
data_range_mode	FirstNonBlank ExplicitRange	Indique le mode de détermination de l'intervalle.
blank_rows	StopReading ReturnBlankRows	Quand la propriété data_range_mode est paramétrée sur <i>PremièreLigneRenseignée</i> , cette propriété indique comment traiter les lignes non renseignées.
explicit_range_start	chaîne	Quand la propriété data_range_mode est paramétrée sur <i>IntervalleExplicite</i> , cette propriété indique le point de départ de l'intervalle à lire.
explicit_range_end	chaîne	
read_field_names	Booléen	Indique si la première ligne de l'intervalle spécifié doit être utilisée pour les noms de champ (de colonne).

Propriétés de evimportnode



Le nœud Enterprise View crée une connexion à un IBM SPSS Collaboration and Deployment Services Repository, vous permettant de lire des données Enterprise View dans un flux et de regrouper un modèle dans un scénario accessible depuis le référentiel par d'autres utilisateurs. [Pour plus d'informations, reportez-vous à la section Noeud Enterprise View dans le chapitre 2 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create evimportnode
set :evimportnode.connection = ['Training data', '/Application views/Marketing', 'LATEST', 'Analytic',
'/Data Providers/Marketing']
```

```
set :evimportnode.tablename = "cust1"
```

propriétés evimportnode	Le type de données	Description de la propriété
connection	Liste	Propriété structurée : liste de paramètres constituant une connexion Enterprise View. Syntaxe : evimportnode.connection = [description,app_view_path, app_view_version_label,environment,DPD_path]
tablename	chaîne	Nom d'un tableau dans la Vue d'application.

Propriétés de fixedfilenode



Le nœud Fixe permet d'importer les données de fichiers texte de longueur fixe, c'est-à-dire les fichiers dont les champs ne sont pas délimités, mais commencent au même endroit et sont de longueur fixe. Les données générées automatiquement ou héritées sont souvent stockées au format de longueur fixe. [Pour plus d'informations, reportez-vous à la section Noeud Fixe dans le chapitre 2 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create fixedfilenode
set :fixedfilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :fixedfilenode.record_len = 32
set :fixedfilenode.skip_header = 1
set :fixedfilenode.fields = [{'Age' 1 3} {'Sex' 5 7} {'BP' 9 10} {'Cholesterol' 12 22} {'Na' 24 25} {'K' 27 27} {'Drug' 29 32}]
set :fixedfilenode.decimal_symbol = Period
set :fixedfilenode.lines_to_scan = 30
```

propriétés fixedfilenode	Le type de données	Description de la propriété
record_len	nombre	Indique le nombre de caractères dans chaque enregistrement.
line_oriented	booléen	Ignore le caractère de nouvelle ligne figurant à la fin de chaque enregistrement.
decimal_symbol	Default Comma Period	Type de séparateur décimal utilisé dans votre source de données. Exemple : set :fixedfilenode.decimal_symbol = Period
skip_header	nombre	Indique le nombre de lignes à ignorer au début du premier enregistrement. Utile pour ignorer les en-têtes de colonne.
auto_recognize_datetime	booléen	Spécifie si les dates ou les heures sont automatiquement identifiées dans les données source.
lines_to_scan	nombre	Exemple : set :fixedfilenode.lines_to_scan = 50.
fields	Liste	Propriétés structurées. Syntaxe : fixedfilenode.fields = {{field start length} {field start length}}

propriétés fixedfilenode	Le type de données	Description de la propriété
full_filename	chaîne	Nom complet (répertoire compris) du fichier à lire.
strip_spaces	None Left Right Both	Les espaces situés en début et en fin de chaîne sont supprimés lors de l'importation.
invalid_char_mode	Discard Replace	Supprime les caractères non valides (null, 0 ou tout caractère inexistant dans le codage en cours) de l'entrée de données ou remplace les caractères non valides par le symbole représentant un caractère.
invalid_char_replacement	chaîne	
use_custom_values	booléen	Propriété saisie avec la syntaxe suivante : set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	Cette propriété est applicable uniquement si un stockage personnalisé est indiqué. Exemple : set:varfilenode.custom Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_date_format.'LaunchDate' = "DDMMYY"

propriétés fixedfilenode	Le type de données	Description de la propriété
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Cette propriété est applicable uniquement si un stockage personnalisé est indiqué. Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_time_format. 'Initialize' = "HHMM"
custom_decimal_symbol	<i>champ</i>	Applicable uniquement si un stockage personnalisé est indiqué. Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	Indique la méthode de codage de texte.

Propriétés de sasimportnode



Le nœud SAS permet d'importer des données SAS dans IBM® SPSS® Modeler. Pour plus d'informations, reportez-vous à la section Nœud source SAS dans le chapitre 2 dans *Nœuds source, exécution et de sortie de IBM SPSS Modeler 15*.

Exemple

```
create sasimportnode
set :sasimportnode.format = Windows
set :sasimportnode.full_filename = "C:/data/retail.sas7bdat"
set :sasimportnode.member_name = "Test"
set :sasimportnode.read_formats = False
set :sasimportnode.full_format_filename = "Test"
set :sasimportnode.import_names = True
```

propriétés sasimportnode	Le type de données	Description de la propriété
format	Windows UNIX Transport SAS7 SAS8 SAS9	Format du fichier à importer.
full_filename	<i>chaîne</i>	Nom de fichier complet que vous entrez, y compris son chemin d'accès.
member_name	<i>chaîne</i>	Indique le membre à importer du fichier Transport SAS spécifié.

propriétés sasimportnode	Le type de données	Description de la propriété
read_formats	booléen	Lit les formats de données (tels que les étiquettes de variable) du fichier de format spécifié.
full_format_filename	chaîne	
import_names	NamesAndLabels LabelsasNames	Spécifie la méthode de mappage des noms de variable et des étiquettes lors de l'importation.

Propriétés de statisticsimportnode



Le nœud IBM® SPSS® Statistics lit les données du format de fichier *.sav* utilisé par SPSS Statistics ainsi que des fichiers cache enregistrés dans IBM® SPSS® Modeler, qui utilisent le même format. [Pour plus d'informations, reportez-vous à la section Nœud Statistics dans le chapitre 8 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Les propriétés de ce nœud sont décrites dans [Propriétés de statisticsimportnode](#) sur p. 316.

Propriétés de userinputnode



Le nœud Utilisateur représente une façon simple de créer des données synthétiques (à partir de zéro ou en modifiant des données existantes). Ceci est utile, par exemple, si vous souhaitez créer un ensemble de données de test pour la modélisation. [Pour plus d'informations, reportez-vous à la section Nœud Utilisateur dans le chapitre 2 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create userinputnode
set :userinputnode.data.test1 = "2, 4, 8"
set :userinputnode.names = [test1 test2]
set :userinputnode.custom_storage.test1 = Integer
set :userinputnode.data_mode = "Ordered"
```

propriétés userinputnode	Le type de données	Description de la propriété
data		Propriété saisie avec la syntaxe suivante : set :userinputnode.data.Age = "1 2 3 4" La chaîne peut également indiquer les valeurs de taille faible, élevée et de pas, séparées par des virgules. Exemple : set :userinputnode.data.Age = "10, 70, 5" Les données de chaque champ peuvent avoir des longueurs différentes, mais doivent correspondre au stockage du champ. ce champ est créé lorsque vous définissez les valeurs d'un champ absent. De plus, lorsque vous définissez les valeurs d'un champ sur une chaîne vide (" "), le champ indiqué est supprimé.
names		Propriété structurée définissant ou renvoyant une liste de noms de champ générés par le nœud.

propriétés userinputnode	Le type de données	Description de la propriété
		Exemple : ['Field1' 'Field2']
custom_storage	Unknown String Integer Real Time Date Timestamp	Propriété saisie définissant ou renvoyant le stockage d'un champ. Exemple : set :userinputnode.custom_storage.'Age' = "Real"
data_mode	Combined Ordered	Si la propriété est définie sur Combined, les enregistrements sont générés pour chaque combinaison de valeurs définies et de valeurs min/max. Le nombre d'enregistrements générés est égal au produit du nombre de valeurs dans chaque champ. Si la propriété est définie sur Ordered, une valeur est copiée à partir des colonnes de chaque enregistrement pour générer une ligne de données. Le nombre d'enregistrements générés est égal au plus grand nombre de valeurs associées à un champ. Tous les champs comportant moins de valeurs de données ont des valeurs nulles.
values		<i>Cette propriété a été remplacée par userinputnode.data et ne doit plus être utilisée.</i>

Propriétés de variablefilenode



Le nœud Délimité lit les données de fichiers texte de longueur variable, c'est-à-dire les fichiers dont les enregistrements contiennent un nombre fixe de champs et un nombre variable de caractères. Ce nœud est également utile pour les fichiers contenant des textes d'en-tête de longueur fixe et certains types d'annotation. [Pour plus d'informations, reportez-vous à la section Nœud Délimité dans le chapitre 2 dans Nœuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.read_field_names = True
set :variablefilenode.delimit_other = True
set :variablefilenode.other = ','
set :variablefilenode.quotes_1 = Discard
set :variablefilenode.decimal_symbol = Comma
set :variablefilenode.invalid_char_mode = "Replace"
set :variablefilenode.invalid_char_replacement = ""
set :variablefilenode.use_custom_values.Age = True
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
```



```
set :variablefilenode.values.Age = [1 100]
```

propriétés variablefilenode	Le type de données	Description de la propriété
skip_header	<i>nombre</i>	Indique le nombre de caractères à ignorer au début du premier enregistrement. Syntaxe : variablefilenode:skip_header = 3
num_fields_auto	<i>booléen</i>	Détermine automatiquement le nombre de champs dans chaque enregistrement. Les enregistrements doivent être terminés par un caractère de retour à la ligne. Syntaxe : variablefilenode:num_fields_auto
num_fields	<i>nombre</i>	Indiquez manuellement le nombre de champs dans chaque enregistrement.
delimit_space	<i>booléen</i>	Indique le caractère utilisé pour délimiter les champs dans le fichier.
delimit_tab	<i>booléen</i>	
delimit_new_line	<i>booléen</i>	
delimit_non_printing	<i>booléen</i>	
delimit_comma	<i>booléen</i>	Si la virgule sert à la fois de séparateur de champs et de séparateur décimal pour les flux, paramétrez <i>delimit_other</i> sur <i>true</i> , puis indiquez la virgule comme séparateur à l'aide de la propriété <i>other</i> .
delimit_other	<i>booléen</i>	Vous permet d'indiquer un séparateur personnalisé à l'aide de la propriété <i>other</i> .
other	<i>chaîne</i>	Indique le séparateur utilisé quand la propriété <i>delimit_other</i> est paramétrée sur <i>true</i> (vrai).
decimal_symbol	Default Comma Period	Indique le séparateur décimal utilisé dans la source de données.
multi_blank	<i>booléen</i>	Considère plusieurs séparateurs non renseignés non adjacents comme un seul séparateur.
read_field_names	<i>booléen</i>	Considère la première ligne du fichier de données comme des étiquettes de colonne.
strip_spaces	None Left Right Both	Les espaces situés en début et en fin de chaîne sont supprimés lors de l'importation.
invalid_char_mode	Discard Replace	Supprime les caractères non valides (null, 0 ou tout caractère inexistant dans le codage en cours) de l'entrée de données ou remplace les caractères non valides par le symbole représentant un caractère.
invalid_char_replacement	<i>chaîne</i>	
lines_to_scan	<i>nombre</i>	Indique le nombre de lignes à analyser pour les types de données spécifiés.
auto_recognize_datetime	<i>booléen</i>	Spécifie si les dates ou les heures sont automatiquement identifiées dans les données source.

propriétés variablefilenode	Le type de données	Description de la propriété
quotes_1	Discard PairAndDiscard IncludeAsText	Indique la manière dont les guillemets simples sont traités lors de l'importation.
quotes_2	Discard PairAndDiscard IncludeAsText	Indique la manière dont les guillemets doubles sont traités lors de l'importation.
full_filename	chaîne	Nom complet (répertoire compris) du fichier à lire.
use_custom_values	booléen	Propriété saisie avec la syntaxe suivante : set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	Applicable uniquement si un stockage personnalisé est indiqué. Exemple : set:varfilenode.custom Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_date_format. 'LaunchDate' = "DDMMYY"

propriétés variablefilenode	Le type de données	Description de la propriété
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Applicable uniquement si un stockage personnalisé est indiqué. Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_time_format. 'Initialize' = "HHMM"
custom_decimal_symbol	champ	Applicable uniquement si un stockage personnalisé est indiqué. Propriété saisie avec la syntaxe suivante : set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	Indique la méthode de codage de texte.

Propriétés de xmlimportnode



Le nœud source XML importe des données au format XML dans le flux. Vous pouvez importer un fichier ou tous les fichiers dans un répertoire. Vous pouvez aussi spécifier un fichier de schéma à partir duquel lire la structure XML. [Pour plus d'informations, reportez-vous à la section Nœud source XML dans le chapitre 2 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create xmlimportnode
set :xmlimportnode.full_filename = "c:\import\lebooks.xml"
set :xmlimportnode.records = "/author/name"
```

propriétés xmlimportnode	Le type de données	Description de la propriété
read	single directory	Lit un seul fichier de données (par défaut), ou tous les fichiers XML d'un répertoire.
recurse	booléen	Spécifie s'il faut lire des fichiers XML supplémentaires dans tous les sous-répertoires du répertoire spécifié.
full_filename	chaîne	(requis) Chemin complet et nom de fichier du fichier XML à importer (si read = single).
directory_name	chaîne	(requis) Chemin complet et nom du répertoire à partir duquel importer les fichiers XML (si read = directory).

propriétés xmlimportnode	Le type de données	Description de la propriété
full_schema_filename	<i>chaîne</i>	Chemin complet et nom du fichier XSD ou DTD à partir duquel la structure XML est lue. Si vous omettez ce paramètre, la structure est lue à partir du fichier source XML.
records	<i>chaîne</i>	Expression XPath (par exemple. /auteur/nom) pour définir la limite de l'enregistrement. À chaque fois que cet élément est rencontré dans le fichier source, un nouvel enregistrement est créé.
mode	read specify	Lire toutes les données (par défaut), ou spécifier les éléments à lire.
fields		Liste des éléments (éléments et attributs) à importer. Chaque élément de la liste est une expression XPath.

Propriétés des noeuds d'opérations sur les lignes

Propriétés de appendnode



Le noeud Ajouter réalise la concaténation d'ensembles d'enregistrements. Il permet de combiner des ensembles de données dont les structures sont similaires, mais les données différentes. [Pour plus d'informations, reportez-vous à la section Noeud Ajouter dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create appendnode
set :appendnode.match_by = Name
set :appendnode.match_case = True
set :appendnode.include_fields_from = All
set :appendnode.create_tag_field = True
set :appendnode.tag_field_name = "Append_Flag"
```

propriétés appendnode	Le type de données	Description de la propriété
match_by	Position Name	Vous pouvez ajouter des ensembles de données sur la base de la position des champs dans la source de données principale, ou du nom des champs dans les ensembles de données d'entrée.
match_case	booléen	Active la distinction des majuscules/minuscules lors de la mise en correspondance des noms de champ.
include_fields_from	Main All	
create_tag_field	booléen	
tag_field_name	chaîne	

Propriétés de aggregatenode



Le noeud Agréger remplace une séquence d'enregistrements d'entrée par des enregistrements de sortie abrégés et agrégés. [Pour plus d'informations, reportez-vous à la section Noeud Agréger dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create aggregatenode
connect :databasenode to :aggregatenode
set :aggregatenode.contiguous = True
```

```

set :aggreatenode.keys = ['Drug']
set :aggreatenode.aggregates.Age = [Sum Mean]
set :aggreatenode.inc_record_count = True
set :aggreatenode.count_field = "index"
set :aggreatenode.extension = "Aggregated_"
set :aggreatenode.add_as = Prefix

```

propriétés aggreatenode	Le type de données	Description de la propriété
keys	<i>[champ champ ... champ]</i>	Affiche les champs qui peuvent être utilisés comme clés pour l'agrégation. Par exemple, si les champs Sex et Région sont vos champs-clés, chaque combinaison unique de M et F avec les régions N et S (quatre combinaisons uniques) aura un enregistrement agrégé.
contiguous	<i>booléen</i>	Sélectionnez cette option si vous savez que tous les enregistrements présentant les mêmes valeurs clés sont groupés dans l'entrée (par exemple, si l'entrée est triée dans les champs clés). Ainsi, vous améliorez les performances.
aggregates		Propriété structurée affichant les champs numériques dont les valeurs seront agrégées, ainsi que les modes d'agrégation sélectionnés. Exemple : set :aggreatenode. aggregates.Age = [Sum Mean Min Max SDev Median Count Variance Firstquartile Thirdquartile], où les méthodes d'agrégation désirées sont incluses dans la liste.
extension	<i>chaîne</i>	Indique un préfixe ou un suffixe pour les champs agrégés dupliqués (exemple ci-dessous).
add_as	Suffix Prefix	
inc_record_count	<i>booléen</i>	Crée un champ supplémentaire qui indique le nombre d'enregistrements d'entrée agrégés pour constituer chaque enregistrement agrégé.
count_field	<i>chaîne</i>	Indique le nom du champ de nombre d'enregistrements.

Propriétés de **balancenode**



Le noeud **Equilibrer** corrige les déséquilibres survenant dans un ensemble de données, de manière à respecter une condition précise. La règle d'équilibrage ajuste la proportion d'enregistrements présentant une condition **True** (vrai) par rapport au facteur indiqué. [Pour plus d'informations, reportez-vous à la section Noeud Equilibrer dans le chapitre 3 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create balancenode
set :balancenode.training_data_only = true
set :balancenode.directives = \
  [{1.3 "Age > 60"}{1.5 "Na > 0.5"}]
```

propriétés balancenode	Le type de données	Description de la propriété
directives		Propriété structurée permettant d'équilibrer la proportion de valeurs de champ en fonction du nombre spécifié (reportez-vous à l'exemple ci-dessous).
training_data_only	booléen	Spécifie que seules des données d'apprentissage devraient être équilibrées. Si aucun champ de partitionnement n'est présent dans le flux, cette option n'est pas prise en compte.

Exemple

```
create balancenode
set :balancenode.directives = \
  [{1.3 "Age > 60"}{1.5 "Na > 0.5"}]
```

Cette propriété de noeud utilise le format suivant :

[{ chaîne de nombres } \ { chaîne de nombres } \ ... { chaîne de nombres }].

Remarque : Si des chaînes (utilisant des guillemets doubles) sont intégrées à l'expression, elles doivent être précédées du caractère d'échappement "\". Le caractère "\" est également le caractère de continuation de ligne, qui vous permet d'aligner les arguments pour plus de clarté.

Propriétés de distinctnode

Le noeud Distinguer supprime les enregistrements en double, soit en incluant le premier enregistrement dans le flux de données, soit en le supprimant et en incluant ses doublons dans le flux de données. [Pour plus d'informations, reportez-vous à la section Noeud Distinguer dans le chapitre 3 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create distinctnode
set :distinctnode.mode = Include
set :distinctnode.fields = ['Age' 'Sex']
set :distinctnode.keys_pre_sorted = True
```

propriétés distinctnode	Le type de données	Description de la propriété
mode	Include Discard	Vous pouvez enlever le premier enregistrement distinct du flux de données, ou l'isoler et transmettre les éventuels enregistrements dupliqués au flux de données.
fields	[<i>champ champ champ</i>]	Affiche les champs utilisés pour détecter les enregistrements identiques.
low_distinct_key_count	booléen	Spécifie que vous avez seulement un petit nombre d'enregistrements et/ou un petit nombre de valeurs uniques du/des champ(s)-clé(s).
keys_pre_sorted	booléen	Spécifie que tous les enregistrements avec les mêmes valeurs-clés sont regroupés dans l'entrée.

Propriétés de mergenode



Le noeud Fusionner permet de créer, à partir de plusieurs enregistrements d'entrée, un seul enregistrement de sortie contenant tout ou partie des champs d'entrée. Il sert notamment à fusionner des données provenant de différentes sources, telles que les données client internes et les données démographiques acquises. [Pour plus d'informations](#), reportez-vous à la section Noeud Fusionner dans le chapitre 3 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.

Exemple

```
create mergenode
connect customerdata to :mergenode
connect salesdata to :mergenode
set :mergenode.method = Keys
set :mergenode.key_fields = ['id']
set :mergenode.common_keys = true
set :mergenode.join = PartialOuter
set :mergenode.outer_join_tag.2 = true
set :mergenode.outer_join_tag.4 = true
set :mergenode.single_large_input = true
set :mergenode.single_large_input_tag = '2'
set :mergenode.use_existing_sort_keys = true
```



```
set :mergenode.existing_sort_keys = [{'id' Ascending}]
```

propriétés mergenode	Le type de données	Description de la propriété
method	Order Keys Condition	Indique si les enregistrements font l'objet d'une fusion dans l'ordre dans lequel ils sont répertoriés dans les fichiers de données, si des champs-clés sont utilisés pour fusionner les enregistrements comportant des valeurs identiques dans ces champs-clés, ou si des enregistrements feront l'objet d'une fusion si une condition particulière est satisfaite.
condition	chaîne	Si method est défini sur Condition, spécifie la condition d'inclusion ou d'exclusion des enregistrements.
key_fields	[<i>champ champ champ</i>]	
common_keys	booléen	
join	Inner FullOuter PartialOuter Anti	Voir l'exemple comme suit : set :merge.join = FullOuter
outer_join_tag.n	booléen	Dans cette propriété, <i>n</i> est le nom de la balise tel qu'il apparaît dans la boîte de dialogue de sélection de l'ensemble de données. Remarque : vous pouvez indiquer plusieurs noms de balise, puisque n'importe quel nombre d'ensembles de données peut générer des enregistrements incomplets.
single_large_input	booléen	Indique si une fonction d'optimisation est utilisée lorsqu'une entrée est plus volumineuse que les autres.
single_large_input_tag	chaîne	Indique le nom de la balise tel qu'il apparaît dans la boîte de dialogue de sélection de l'ensemble de données volumineux. L'utilisation de cette propriété diffère légèrement de celle de la propriété outer_join_tag (booléen/chaîne) car un seul ensemble de données d'entrée peut être spécifié.
use_existing_sort_keys	booléen	Indique si les entrées sont déjà triées en fonction d'un ou de plusieurs champs-clés.
existing_sort_keys	[{chaîne Ascending} \ {chaîne Descending}]	Indique les champs déjà triés ainsi que le sens du tri.

Propriétés de rfmaggregatenode



Le noeud agrégé Recency, Frequency, Monetary (RFM) vous permet de prendre les données de l'historique des transactions d'un client, d'en éliminer les éventuelles données inutilisées et de combiner le reste des données de transaction sur une seule ligne qui indique la date de la dernière consultation, le nombre de transactions réalisées et la valeur monétaire totale de ces transactions. [Pour plus d'informations, reportez-vous à la section Noeud Agréger RFM dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```

create rfmaggregatenode
connect :fillernode to :rfmaggregatenode
set :rfmaggregatenode.relative_to = Fixed
set :rfmaggregatenode.reference_date = "2007-10-12"
set :rfmaggregatenode.id_field = "CardID"
set :rfmaggregatenode.date_field = "Date"
set :rfmaggregatenode.value_field = "Amount"
set :rfmaggregatenode.only_recent_transactions = True
set :rfmaggregatenode.transaction_date_after = "2000-10-01"

```

propriétés rfmaggregatenode	Le type de données	Description de la propriété
relative_to	Fixed Today	Spécifie la date à partir de laquelle la récence des transactions sera calculée.
reference_date	Date	Disponible uniquement si Fixed est défini sur relative_to.
contiguous	booléen	Si vos données sont pré-triées de façon à ce que tous les enregistrements avec le même ID apparaissent ensemble dans le flux de données, la sélection de cette option accélère le traitement.
id_field	champ	Spécifie le champ à utiliser pour identifier le client et ses transactions.
date_field	champ	Spécifie le champ de date à utiliser pour calculer la récence.
value_field	champ	Spécifie le champ à utiliser pour calculer la valeur monétaire.
extension	chaîne	Indique un préfixe ou un suffixe pour les champs agrégés en double.
add_as	Suffix Prefix	Spécifie si l'extension doit être ajoutée comme suffixe ou comme préfixe.
discard_low_value_records	booléen	Active l'utilisation du paramètre discard_records_below.
discard_records_below	nombre	Spécifie une valeur minimale sous laquelle les éventuels détails de la transaction ne seront pas utilisés pour calculer les totaux RFM. L'unité de valeur fait référence au champ value sélectionné.
only_recent_transactions	booléen	Permet l'utilisation des paramètres specify_transaction_date ou transaction_within_last.
specify_transaction_date	booléen	
transaction_date_after	Date	Disponible uniquement si specify_transaction_date est sélectionné. Spécifie la date de transaction après laquelle les enregistrements seront inclus dans votre analyse.

propriétés rfmaggregatenode	Le type de données	Description de la propriété
transaction_within_last	nombre	Disponible uniquement si transaction_within_last est sélectionné. Spécifie le nombre et le type de périodes (jours, semaines, mois ou années) provenant de la valeur de calcul de récence par rapport à la date après laquelle les enregistrements seront inclus dans votre analyse.
transaction_scale	Days Weeks Months Years	Disponible uniquement si transaction_within_last est sélectionné. Spécifie le nombre et le type de périodes (jours, semaines, mois ou années) provenant de la valeur de calcul de récence par rapport à la date après laquelle les enregistrements seront inclus dans votre analyse.
save_r2	booléen	Affiche la date de l'avant-dernière transaction pour chaque client.
save_r3	booléen	Disponible uniquement si save_r2 est sélectionné. Affiche la date de la troisième transaction la plus récente pour chaque client.

Propriétés de samplénodé



Le noeud Echantillonner sélectionne un sous-ensemble d'enregistrements. Divers types d'échantillons sont pris en charge, notamment les échantillons stratifiés, en classe et non aléatoires (structurés). L'échantillonnage peut être utile pour améliorer les performances et pour sélectionner des groupes d'enregistrements associés ou des transactions pour analyse. [Pour plus d'informations, reportez-vous à la section Noeud Echantillon dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
/* Create two Sample nodes to extract
different samples from the same data */

create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"

set node = create samplénodé at 300 100
rename ^node as 'First 500'
connect :variablefilenode to 'First 500'
set 'First 500':samplénodé.method = Simple
set 'First 500':samplénodé.mode = Include
set 'First 500':samplénodé.sample_type = First
set 'First 500':samplénodé.first_n = 500

set node = create samplénodé at 300 200
rename ^node as 'Custom Strata'
connect :variablefilenode to 'Custom Strata'
set 'Custom Strata':samplénodé.method = Complex
set 'Custom Strata':samplénodé.stratify_by = ['Sex' 'Cholesterol']
```

```

set 'Custom Strata':samplenode.sample_units = Proportions
set 'Custom Strata':samplenode.sample_size_proportions = Custom
set 'Custom Strata':samplenode.sizes_proportions= \
  [{"M" "High" "Default"}{"M" "Normal" "Default"} \
  {"F" "High" "0.3"}{"F" "Normal" "0.3"}]

```

propriétés samplenode	Le type de données	Description de la propriété
method	Simple Complexe	
mode	Include Discard	Enlève ou isole les enregistrements qui correspondent à la condition indiquée.
sample_type	First OneInN RandomPct	Indique la méthode d'échantillonnage. Voir l'exemple comme suit : set :samplenode.sample_type = First set :samplenode.first_n = 100
first_n	entier	Les enregistrements jusqu'au point de césure indiqué sont inclus ou exclus.
one_in_n	nombre	Inclut ou exclut un enregistrement tous les <i>n</i> enregistrements.
rand_pct	nombre	Indique le pourcentage d'enregistrements à inclure ou à exclure.
use_max_size	booléen	Active l'utilisation du paramètre maximum_size.
maximum_size	entier	Indiquez la taille maximale des échantillons à inclure dans le flux de données ou à exclure du flux. Cette option est redondante et par conséquent désactivée si les options First et Include sont sélectionnées.
set_random_seed	booléen	Active l'utilisation du paramètre de graine aléatoire.
random_seed	entier	Indique la valeur utilisée en tant que graine aléatoire.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	nombre	
fixed_counts	entier	
variable_proportions	champ	
variable_counts	champ	
use_min_stratum_size	booléen	
minimum_stratum_size	entier	Cette option s'applique uniquement lorsqu'un échantillon Complexe est utilisé avec Sample units=Proportions.
use_max_stratum_size	booléen	

propriétés samplenode	Le type de données	Description de la propriété
maximum_stratum_size	entier	Cette option s'applique uniquement lorsqu'un échantillon Complexe est utilisé avec Sample units=Proportions.
clusters	champ	
stratify_by	[champ1 ... champN]	
specify_input_weight	booléen	
input_weight	champ	
new_output_weight	chaîne	
sizes_proportions	[{stringvaleur de chaîne} {stringvaleur de chaîne} ...]	Si sample_units=proportions et sample_size_proportions=Custom, spécifie une valeur pour chaque combinaison de valeurs possible pour les champs de stratification.
default_proportion	nombre	
sizes_counts	[{stringvaleur de chaîne} {stringvaleur de chaîne} ...]	Spécifie une valeur pour chaque combinaison de valeurs possible pour les champs de stratification. L'utilisation est similaire à celle de sizes_proportions, mais en spécifiant un entier au lieu d'une proportion.
default_count	nombre	

Propriétés de **selectnode**



Le noeud Sélectionner permet de sélectionner ou d'exclure des sous-ensembles d'enregistrements d'un flux de données sur la base d'une condition spécifique. Par exemple, vous pouvez sélectionner les enregistrements qui appartiennent à un secteur de ventes particulier. [Pour plus d'informations, reportez-vous à la section Noeud Sélectionner dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create selectnode
set :selectnode.mode = Include
set :selectnode.condition = "Age < 18"
```

propriétés selectnode	Le type de données	Description de la propriété
mode	Include Discard	Indique si les enregistrements sélectionnés sont à enlever ou à isoler.
condition	chaîne	Condition d'inclusion ou d'exclusion des enregistrements.

Propriétés de **sortnode**



Le noeud Trier trie les enregistrements par ordre croissant ou décroissant, en fonction de la valeur d'un ou de plusieurs champs. [Pour plus d'informations, reportez-vous à la section Noeud Trier dans le chapitre 3 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```

create sortnode
set :sortnode.keys = [{'Age' Ascending}{'Sexe' Descending}]
set :sortnode.default_ascending = False
set :sortnode.use_existing_keys = True
set :sortnode.existing_keys = [{'Age' Ascending}]

```

propriétés sortnode	Le type de données	Description de la propriété
keys	[{chaîne Ascending} \ {chaîne Descending}]	Spécifie les champs servant de base au tri (exemple ci-dessous). Si aucune direction n'est indiquée, la valeur utilisée est celle par défaut.
default_ascending	booléen	Indique l'ordre de tri par défaut.
use_existing_keys	booléen	Indique si le tri est optimisé à l'aide de l'ordre de tri précédent des champs déjà triés.
existing_keys		Indique les champs déjà triés ainsi que le sens du tri. Utilise le même format que la propriété keys.

Propriétés des nœuds d'opérations sur les champs

Propriétés de anonymizenode



Le nœud Anonymiser transforme la façon dont les noms et les valeurs des champs sont représentés en aval, masquant ainsi les données d'origine. Cela peut s'avérer utile si vous souhaitez permettre à d'autres utilisateurs de générer des modèles utilisant des données confidentielles, par exemple des noms de clients ou autre. [Pour plus d'informations, reportez-vous à la section Nœud Anonymiser dans le chapitre 4 dans Nœuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create anonymizenode
set: anonymizenode.enable_anonymize = âge
set: anonymizenode.use_prefix = true
set: anonymizenode.prefix = "myprefix"
set: anonymizenode.transformation = Random
set: anonymizenode.set_random_seed = true
set: anonymizenode.random_seed = "123"
```

propriétés anonymizenode	Le type de données	Description de la propriété
enable_anonymize	<i>booléen</i>	Lorsque cette propriété est définie sur T, elle active l'anonymisation des valeurs de champ (revient à sélectionner Oui pour ce champ dans la colonne Anonymiser des valeurs).
use_prefix	<i>booléen</i>	Lorsque cette propriété est définie sur T, un préfixe personnalisé sera utilisé, s'il en existe un. S'applique aux champs qui seront anonymisés grâce à la méthode de hachage et revient à sélectionner la case d'option Personnalisé dans la boîte de dialogue Remplacer les valeurs pour ce champ.
prefix	<i>chaîne</i>	Revient à taper un préfixe dans la zone de texte de la boîte de dialogue Remplacer les valeurs. Le préfixe par défaut est la valeur par défaut si rien d'autre n'a été indiqué.
transformation	Random Fixed	Détermine si les paramètres de transformation pour un champ anonymisé par la méthode de transformation seront aléatoires ou fixes.
set_random_seed	<i>booléen</i>	Lorsque cette propriété est définie sur T, la valeur de graine indiquée sera utilisée (si la propriété transformation est également définie sur Aléatoire).
random_seed	<i>entier</i>	Lorsque set_random_seed est défini sur T, il s'agit de la graine pour le nombre aléatoire.

propriétés anonymizenode	Le type de données	Description de la propriété
scale	nombre	Lorsque la propriété transformation est définie sur Fixe, cette valeur sert pour la mise à l'échelle. La valeur d'échelle maximale est normalement 10, mais elle peut être diminuée pour éviter tout dépassement.
translate	nombre	Lorsque la propriété transformation est définie sur Fixe, cette valeur sert à la « traduction ». La valeur maximale de conversion est normalement 1 000, mais elle peut être diminuée pour éviter tout dépassement.

Propriétés autodatapreinode



Le noeud de préparation automatisée de données (ADP) peut analyser vos données, identifier des corrections et filtrer des champs qui sont problématiques ou qui sont peu susceptibles d'être utiles. Il peut aussi créer de nouveaux attributs le cas échéant et améliorer la performance au moyen de techniques de filtrage et d'échantillonnage intelligentes. Vous pouvez utiliser le noeud de manière totalement automatisée, en laissant le noeud choisir et appliquer les corrections, ou vous pouvez prévisualiser les modifications avant qu'elles ne soient mises en place et les accepter, les rejeter ou les modifier selon les besoins. [Pour plus d'informations, reportez-vous à la section Préparation automatique des données dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create autodatapreinode
set: autodatapreinode.objective = Balanced
set: autodatapreinode.excluded_fields = Filter
set: autodatapreinode.prepare_dates_and_times = true
set: autodatapreinode.compute_time_until_date = true
set: autodatapreinode.reference_date = Today
set: autodatapreinode.units_for_date_durations = Automatic
```

propriétés autodatapreinode	Le type de données	Description de la propriété
objective	Balanced Speed Accuracy Custom	
custom_fields	booléen	Si la valeur est définie sur true (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur false (faux), les paramètres actuels provenant d'un noeud Typer en amont sont utilisés.
target	champ	Spécifie un champ cible unique.
inputs	[champ1 ... champN]	Champs d'entrée ou variable indépendante utilisés par le modèle.
use_frequency	booléen	
frequency_field	champ	
use_weight	booléen	

propriétés autodatapreinode	Le type de données	Description de la propriété
weight_field	<i>champ</i>	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	<i>booléen</i>	Contrôle l'accès à tous les champs date et heure
compute_time_until_date	<i>booléen</i>	
reference_date	Today Fixed	
fixed_date	<i>Date</i>	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	<i>booléen</i>	
reference_time	CurrentTime Fixed	
fixed_time	<i>heure</i>	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	<i>booléen</i>	
extract_month_from_date	<i>booléen</i>	
extract_day_from_date	<i>booléen</i>	
extract_hour_from_time	<i>booléen</i>	
extract_minute_from_time	<i>booléen</i>	
extract_second_from_time	<i>booléen</i>	
exclude_low_quality_inputs	<i>booléen</i>	
exclude_too_many_missing	<i>booléen</i>	
maximum_percentage_missing	<i>nombre</i>	
exclude_too_many_categories	<i>booléen</i>	
maximum_number_categories	<i>nombre</i>	
exclude_if_large_category	<i>booléen</i>	
maximum_percentage_category	<i>nombre</i>	
prepare_inputs_and_target	<i>booléen</i>	
adjust_type_inputs	<i>booléen</i>	
adjust_type_target	<i>booléen</i>	
reorder_nominal_inputs	<i>booléen</i>	
reorder_nominal_target	<i>booléen</i>	
replace_outliers_inputs	<i>booléen</i>	
replace_outliers_target	<i>booléen</i>	

propriétés autodatapreinode	Le type de données	Description de la propriété
replace_missing_continuous_inputs	<i>booléen</i>	
replace_missing_continuous_target	<i>booléen</i>	
replace_missing_nominal_inputs	<i>booléen</i>	
replace_missing_nominal_target	<i>booléen</i>	
replace_missing_ordinal_inputs	<i>booléen</i>	
replace_missing_ordinal_target	<i>booléen</i>	
maximum_values_for_ordinal	<i>nombre</i>	
minimum_values_for_continuous	<i>nombre</i>	
outlier_cutoff_value	<i>nombre</i>	
outlier_method	Replace Delete	
rescale_continuous_inputs	<i>booléen</i>	
rescaling_method	MinMax ZScore	
min_max_minimum	<i>nombre</i>	
min_max_maximum	<i>nombre</i>	
z_score_final_mean	<i>nombre</i>	
z_score_final_sd	<i>nombre</i>	
rescale_continuous_target	<i>booléen</i>	
target_final_mean	<i>nombre</i>	
target_final_sd	<i>nombre</i>	
transform_select_input_fields	<i>booléen</i>	
maximize_association_with_target	<i>booléen</i>	
p_value_for_merging	<i>nombre</i>	
merge_ordinal_features	<i>booléen</i>	
merge_nominal_features	<i>booléen</i>	
minimum_cases_in_category	<i>nombre</i>	
bin_continuous_fields	<i>booléen</i>	
p_value_for_binning	<i>nombre</i>	
perform_feature_selection	<i>booléen</i>	
p_value_for_selection	<i>nombre</i>	
perform_feature_construction	<i>booléen</i>	
transformed_target_name_extension	<i>chaîne</i>	
transformed_inputs_name_extension	<i>chaîne</i>	
constructed_features_root_name	<i>chaîne</i>	
years_duration_name_extension	<i>chaîne</i>	
months_duration_name_extension	<i>chaîne</i>	
days_duration_name_extension	<i>chaîne</i>	
hours_duration_name_extension	<i>chaîne</i>	

propriétés autodatapreinode	Le type de données	Description de la propriété
minutes_duration_name_extension	chaîne	
seconds_duration_name_extension	chaîne	
year_cyclical_name_extension	chaîne	
month_cyclical_name_extension	chaîne	
day_cyclical_name_extension	chaîne	
hour_cyclical_name_extension	chaîne	
minute_cyclical_name_extension	chaîne	
second_cyclical_name_extension	chaîne	

Propriétés de binningnode



Le nœud Discrétiser crée automatiquement des champs nominaux (ensemble) sur la base des valeurs d'un ou de plusieurs champs continus (intervalle numérique) existants. Par exemple, vous pouvez transformer un champ continu de revenus en un nouveau champ catégoriel contenant des groupes de revenus comme écarts par rapport à la moyenne. Une fois les intervalles du nouveau champ créés, vous pouvez générer un nœud Calculer à partir des points de césure. [Pour plus d'informations, reportez-vous à la section Nœud Discrétiser dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create binningnode
set :binningnode.fields = [Na K]
set :binningnode.method = Rank
set :binningnode.fixed_width_name_extension = "_binned"
set :binningnode.fixed_width_add_as = Suffix
set :binningnode.fixed_bin_method = Count
set :binningnode.fixed_bin_count = 10
set :binningnode.fixed_bin_width = 3.5
set :binningnode.tile10 = true
```

propriétés binningnode	Le type de données	Description de la propriété
fields	[champ1 champ2 ... champn]	Champs continus (intervalle numérique) en attente de transformation. Vous pouvez créer des intervalles pour plusieurs champs simultanément.
method	FixedWidth EqualCount Rank SDev Optimal	Méthode utilisée pour déterminer les points de césure des nouveaux intervalles de champ (catégories).
recalculate_bins	Always IfNecessary	Indique si les intervalles sont recalculés et les données placées dans l'intervalle approprié à chaque exécution du nœud ou si les données sont uniquement ajoutées aux intervalles existants et aux nouveaux intervalles éventuellement ajoutés.

propriétés binningnode	Le type de données	Description de la propriété
fixed_width_name_extension	chaîne	L'extension par défaut est <i>_BIN</i> .
fixed_width_add_as	Suffix Prefix	Indique si l'extension est ajoutée à la fin (suffixe) ou au début (préfixe) du nom du champ. L'extension par défaut est <i>income_BIN</i> .
fixed_bin_method	Width Count	
fixed_bin_count	entier	Désigne l'entier déterminant le nombre d'intervalles de largeur fixe (catégories) des nouveaux champs.
fixed_bin_width	réel	Valeur (entier ou réel) permettant de calculer la largeur de l'intervalle.
equal_count_name_extension	chaîne	L'extension par défaut est <i>_TILE</i> .
equal_count_add_as	Suffix Prefix	Indique une extension, suffixe ou préfixe, utilisée pour le nom de champ généré à l'aide de centiles standard. L'extension par défaut est <i>_TILE</i> plus <i>N</i> , <i>N</i> étant le numéro du quantile.
tile4	booléen	Génère quatre intervalles de quintile, chacun contenant 25 % des observations.
tile5	booléen	Génère cinq intervalles de quintiles.
tile10	booléen	Génère 10 intervalles de déciles.
tile20	booléen	Génère 20 intervalles de vingtiles.
tile100	booléen	Génère 100 intervalles de centiles.
use_custom_tile	booléen	
custom_tile_name_extension	chaîne	L'extension par défaut est <i>_TILEN</i> .
custom_tile_add_as	Suffix Prefix	
custom_tile	entier	
equal_count_method	RecordCount ValueSum	La méthode RecordCount vise à affecter un nombre égal d'enregistrements à chaque intervalle, tandis que la méthode ValueSum affecte les enregistrements de sorte que la somme des valeurs de chaque intervalle soit égale.
tied_values_method	Next Current Random	Spécifie l'intervalle dans lequel les données de valeur ex æquo doivent être placées.
rank_order	Ascending Descending	Cette propriété peut avoir la valeur Ascending (la valeur la plus faible est marquée 1) ou Descending (la valeur la plus élevée est marquée 1).
rank_add_as	Suffix Prefix	Cette option s'applique au rang, au rang fractionnaire et au rang de pourcentage.
rank	booléen	
rank_name_extension	chaîne	L'extension par défaut est <i>_RANK</i> .

propriétés binningnode	Le type de données	Description de la propriété
rank_fractional	booléen	Permet de classer les observations dans lesquelles la valeur du nouveau champ équivaut au rang divisé par la somme des pondérations des observations non manquantes. Les rangs fractionnaires sont compris dans l'intervalle 0–1.
rank_fractional_name_extension	chaîne	L'extension par défaut est <i>_F_RANK</i> .
rank_pct	booléen	Chaque rang est divisé par le nombre d'enregistrements avec valeurs valides et multiplié par 100. Les rangs fractionnaires de pourcentage sont compris dans l'intervalle 1–100.
rank_pct_name_extension	chaîne	L'extension par défaut est <i>_P_RANK</i> .
sdev_name_extension	chaîne	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	chaîne	L'extension par défaut est <i>_OPTIMAL</i> .
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	champ	Champ choisi comme champ de superviseur et auquel les champs sélectionnés pour la création d'intervalles sont associés.
optimal_merge_bins	booléen	Indique que tous les intervalles présentant peu d'observations seront ajoutés à un intervalle voisin plus grand.
optimal_small_bin_threshold	entier	
optimal_pre_bin	booléen	Indique que la pré-crédation d'intervalles de l'ensemble de données va avoir lieu.
optimal_max_bins	entier	Définit une limite supérieure afin d'éviter de créer un nombre d'intervalles trop important.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded Bounded	

Propriétés de derivenode



Le nœud Calculer modifie les valeurs de données ou crée des nouveaux champs à partir d'un ou de plusieurs champs existants. Il crée des champs de type formule, booléen, ensemble, nominal, statistiques, comptage et conditionnel. [Pour plus d'informations, reportez-vous à la section Nœud Calculer dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
# Créer et configurer un noeud de champ Calculer booléen
create derivenode
rename derive:derivenode as "Flag"
set Flag:derivenode.new_name = "DrugX_Flag"
set Flag:derivenode.result_type = Flag
set Flag:derivenode.flag_true = 1
set Flag:derivenode.flag_false = 0
set Flag:derivenode.flag_expr = "Drug = X"
# Créer et configurer un noeud de champ Calculer conditionnel
create derivenode
rename derive:derivenode as "Conditional"
set Conditional:derivenode.result_type = Conditional
set Conditional:derivenode.cond_if_cond = "@OFFSET('\Age', 1) = '\Age'"
set Conditional:derivenode.cond_then_expr = "@OFFSET('\Age', 1) = '\Age' >> @INDEX"
set Conditional:derivenode.cond_else_expr = "\Age"
```

propriétés derivenode	Le type de données	Description de la propriété
new_name	chaîne	Nom du nouveau champ.
mode	Single Multiple	Spécifie un ou plusieurs champs.
fields	[<i>champ champ</i> <i>champ</i>]	Utilisé en mode Multiple uniquement pour sélectionner plusieurs champs.
name_extension	chaîne	Indique l'extension des nouveaux noms de champ.
add_as	Suffix Prefix	Ajoute l'extension du nom de champ en tant que préfixe (au début) ou en tant que suffixe (à la fin).
result_type	Formule Flag Set State Count Conditional	Les six types de nouveau champ que vous pouvez créer.
formula_expr	chaîne	Expression de calcul de la valeur du nouveau champ dans un noeud Calculer
flag_expr	chaîne	
flag_true	chaîne	
flag_false	chaîne	
set_default	chaîne	
set_value_cond	chaîne	Propriété structurée (définition de la condition associée à une valeur donnée). Syntaxe : set :derivenode. set_value_cond. Retired = 'age > 65'
state_on_val	chaîne	Indique la valeur du nouveau champ si la condition Activé est vérifiée.
state_off_val	chaîne	Indique la valeur du nouveau champ si la condition Désactivé est vérifiée.
state_on_expression	chaîne	

propriétés derivenode	Le type de données	Description de la propriété
state_off_expression	chaîne	
state_initial	On Off	Affecte à chaque enregistrement du nouveau champ la valeur initiale On ou Off. Cette valeur peut changer au fur et à mesure que les conditions sont respectées.
count_initial_val	chaîne	
count_inc_condition	chaîne	
count_inc_expression	chaîne	
count_reset_condition	chaîne	
cond_if_cond	chaîne	
cond_then_expr	chaîne	
cond_else_expr	chaîne	

Propriétés de **ensemblenode**



Le nœud Ensemble combine deux ou plusieurs nuggets de modèles pour obtenir des prévisions plus précises que celles acquises à partir d'un modèle quelconque. [Pour plus d'informations, reportez-vous à la section Nœud Ensemble dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
# Créer et configurer un nœud Ensemble
# Utilisez ce nœud avec les modèles dans le dossier demos\streams\pm_binaryclassifier.str
create ensemblenode
set :ensemblenode.ensemble_target_field = response
set :ensemblenode.filter_individual_model_output = false
set :ensemblenode.flag_ensemble_method = ConfidenceWeightedVoting
set :ensemblenode.flag_voting_tie_selection = HighestConfidence
```

propriétés ensemblenode	Le type de données	Description de la propriété
ensemble_target_field	champ	Indique le champ cible pour tous les modèles utilisés dans l'ensemble.
filter_individual_model_output	booléen	Indique si les résultats de scoring de modèles individuels doivent être supprimés.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ booléen.

propriétés ensemblenode	Le type de données	Description de la propriété
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ booléen.
set_voting_tie_selection	Random HighestConfidence	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.
calculate_standard_error	<i>booléen</i>	Si le champ cible est continu, un calcul d'erreur standard est exécuté par défaut pour calculer la différence entre les valeurs mesurées ou estimées et les valeurs réelles, et pour montrer la correspondance proche de ces évaluations.

Propriétés de fillernode



Le noeud Remplacer permet de remplacer les valeurs de champ et de modifier le type de stockage. Vous pouvez décider de remplacer les valeurs reposant sur une condition CLEM, telle que @BLANK(@FIELD). Vous pouvez également choisir de remplacer tous les blancs ou toutes les valeurs nulles par une valeur précise. Un noeud Remplacer est souvent associé à un noeud Typer pour remplacer les valeurs manquantes. [Pour plus d'informations, reportez-vous à la section Noeud Remplacer dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create fillernode
set :fillernode.fields = ['Age']
set :fillernode.replace_mode = Always
set :fillernode.condition = "(\ 'Age\ ' > 60) and (\ 'Sex\ ' = \ 'M\ ')"
set :fillernode.replace_with = "\ old man\ "
```

propriétés fillernode	Le type de données	Description de la propriété
fields	[<i>champ champ</i> <i>champ</i>]	Champs de l'ensemble de données dont les valeurs sont examinées et remplacées.
replace_mode	Always Conditional Blank Null BlankAndNull	Vous pouvez remplacer toutes les valeurs, les valeurs vides ou les valeurs nulles, ou effectuer le remplacement en fonction d'une condition définie.
condition	<i>chaîne</i>	
replace_with	<i>chaîne</i>	

Propriétés de filternode



Le nœud Filtrer filtre (supprime) les champs, les renomme et les mappe entre un nœud source et un autre. [Pour plus d'informations, reportez-vous à la section Filtrage ou modification du nom des champs dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create filternode
set :filternode.default_include = True
set :filternode.new_name.'Drug' = 'Chemical'
set :filternode.include.'Drug' = off
```

Utilisation de la propriété default_include. Le paramétrage de la valeur de la propriété default_include n'implique pas l'inclusion ou l'exclusion automatique de tous les champs ; il détermine simplement la valeur par défaut pour la sélection actuelle. Ceci équivaut, d'un point de vue fonctionnel, à cliquer sur le bouton Inclure les champs par défaut dans la boîte de dialogue du nœud Filtrer. Supposons, par exemple, que vous exécutiez le script suivant :

```
set Filter.default_include=False
# Inclure uniquement les champs de la liste
for f in Age Sexe
  set Filter.include.^f=True
endfor
```

Dans ce cas, le nœud transmet les champs *Age* et *Sexe* et ignore les autres. Supposons maintenant que vous exécutiez le même script avec deux autres champs :

```
set Filter.default_include=False
# Inclure uniquement les champs de la liste
for f in TA Na
  set Filter.include.^f=True
endfor
```

Deux champs supplémentaires sont ajoutés au filtre ; par conséquent, quatre champs au total sont transmis (*Age*, *Sexe*, *TA*, *Na*). En d'autres termes, le fait de paramétrer de nouveau la valeur de la propriété default_include sur False (faux) ne réinitialise pas automatiquement tous les champs.

D'autre part, si vous paramétrez maintenant la propriété `default_include` sur `True` (vrai), par le biais d'un script ou de la boîte de dialogue du noeud Filtrer, le résultat est inversé et les quatre champs répertoriés ci-dessus sont ignorés plutôt qu'inclus. Si vous avez un doute, manipulez les commandes de la boîte de dialogue du noeud Filtrer pour comprendre cette interaction.

propriétés <code>filternode</code>	Le type de données	Description de la propriété
<code>default_include</code>	<i>booléen</i>	Propriété saisie permettant d'indiquer si le comportement par défaut consiste à transmettre ou à filtrer les champs : <code>NODE.include.FIELDNAME</code> Voir l'exemple comme suit : <code>set mynode:filternode.default_include = false</code> Le paramétrage de cette propriété n'implique pas l'inclusion ou l'exclusion automatique de tous les champs ; il détermine simplement si les champs sélectionnés sont inclus ou exclus par défaut. Pour consulter d'autres commentaires, reportez-vous à l'exemple ci-dessous.
<code>include</code>	<i>booléen</i>	Propriété saisie (inclusion et suppression de champs). Syntaxe : <code>NODE.include.FIELDNAME</code> Voir l'exemple comme suit : <code>set mynode:filternode.include.Age = false</code>
<code>new_name</code>	<i>chaîne</i>	Voir l'exemple comme suit : <code>set mynode:filternode.new_name.Age = "age"</code>

Propriétés de `historynode`



Le noeud Historiser crée des champs contenant des données provenant de champs d'enregistrements antérieurs. Les noeuds Historiser sont souvent utilisés pour les données séquentielles, telles que les séries temporelles. Avant d'utiliser un noeud Historiser, vous pouvez trier les données à l'aide d'un noeud Trier. [Pour plus d'informations, reportez-vous à la section Noeud Historiser dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create historynode
set :historynode.fields = ['Drug']
set :historynode.offset = 1
set :historynode.span = 3
set :historynode.unavailable = Discard
```

```
set :historynode.fill_with = "undef"
```

propriétés historynode	Le type de données	Description de la propriété
fields	[<i>champ champ champ</i>]	Champs pour lesquels vous souhaitez un historique.
offset	<i>nombre</i>	Indique le dernier enregistrement (avant l'enregistrement actuel) à partir duquel extraire les valeurs de champ historiques.
span	<i>nombre</i>	Indique le nombre d'enregistrements précédents desquels extraire des valeurs.
unavailable	Discard Leave Fill	Propriété destinée à la gestion des enregistrements qui n'ont aucune valeur historique ; fait généralement référence aux premiers enregistrements (situés au début de l'ensemble de données) et pour lesquels il n'existe aucun enregistrement précédent à utiliser en tant qu'historique.
fill_with	String Number	Indique la valeur ou la chaîne à utiliser pour les enregistrements dans lesquels aucune valeur d'historique n'est disponible.

Propriétés de partitionnode



Le nœud Partitionner génère un champ de partition qui répartit les données dans des sous-ensembles distincts pour les étapes d'apprentissage, de test et de validation de la création d'un modèle. [Pour plus d'informations, reportez-vous à la section Nœud Partitionner dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create partitionnode
set :partitionnode.create_validation = True
set :partitionnode.training_size = 33
set :partitionnode.testing_size = 33
set :partitionnode.validation_size = 33
set :partitionnode.set_random_seed = True
set :partitionnode.random_seed = "123"
set :partitionnode.value_mode = System
```

propriétés partitionnode	Le type de données	Description de la propriété
new_name	<i>chaîne</i>	Nom du champ de partition généré par le nœud.
create_validation	<i>booléen</i>	Indique si une partition de validation doit être créée.
training_size	<i>entier</i>	Pourcentage des enregistrements (0 à 100) à allouer à la partition d'apprentissage.
testing_size	<i>entier</i>	Pourcentage des enregistrements (0 à 100) à allouer à la partition de test.

propriétés partitionnode	Le type de données	Description de la propriété
validation_size	entier	Pourcentage des enregistrements (0 à 100) à allouer à la partition de validation. Ignoré si aucune partition de validation n'est créée.
training_label	chaîne	Étiquette de la partition d'apprentissage.
testing_label	chaîne	Étiquette de la partition de test.
validation_label	chaîne	Étiquette de la partition de validation. Ignoré si aucune partition de validation n'est créée.
value_mode	System SystemAndLabel Label	Indique les valeurs utilisées pour représenter chaque partition dans les données. Par exemple, l'échantillon d'apprentissage peut être représenté par l'entier système 1, l'étiquette Training ou une combinaison des deux : 1_Training.
set_random_seed	Booléen	Indique si vous devez utiliser une graine aléatoire définie par l'utilisateur.
random_seed	entier	Valeur de graine aléatoire définie par l'utilisateur. Pour utiliser cette valeur, set_random_seed doit avoir la valeur True.
enable_sql_generation	Booléen	Spécifie s'il faut utiliser la répercussion SQL pour affecter des enregistrements à des partitions.
unique_field		Spécifie le champ d'entrée utilisé pour vérifier que des enregistrements sont attribués à des partitions de manière aléatoire mais répétitive. Pour utiliser cette valeur, enable_sql_generation doit avoir la valeur True.

Propriétés de reclassifynode



Le noeud Recoder permet de transformer un ensemble de valeurs catégorielles en un autre. La recodification est utile pour réduire des catégories ou regrouper des données à analyser. [Pour plus d'informations, reportez-vous à la section Noeud Recoder dans le chapitre 4 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create reclassifynode
set :reclassifynode.mode = Multiple
set :reclassifynode.replace_field = true
set :reclassifynode.field = "Drug"
set :reclassifynode.new_name = "Chemical"
set :reclassifynode.fields = [Drug, BP]
set :reclassifynode.name_extension = "reclassified"
set :reclassifynode.add_as = Prefix
set :reclassifynode.reclassify.'drugA' = 'Yes'
set :reclassifynode.use_default = True
set :reclassifynode.default = "BrandX"
```

```
set :reclassifynode.pick_list = [BrandX, Placebo, Generic]
```

propriétés reclassifynode	Le type de données	Description de la propriété
mode	Single Multiple	Single reclassifie les catégories pour un champ. Multiple active les options permettant de transformer plusieurs champs simultanément.
replace_field	booléen	
field	chaîne	Utilisé en mode Simple uniquement.
new_name	chaîne	Utilisé en mode Simple uniquement.
fields	[champ1 champ2 ... champn]	Utilisé en mode Multiple uniquement.
name_extension	chaîne	Utilisé en mode Multiple uniquement.
add_as	Suffix Prefix	Utilisé en mode Multiple uniquement.
reclassify	chaîne	Propriété structurée (valeur des champs). Syntaxe : NODE.reclassify. OLD_VALUE Voir l'exemple comme suit : set :reclassifynode.reclassify.'drugB' = 'Yes'
use_default	booléen	Utilisez la valeur par défaut.
default	chaîne	Spécifiez une valeur par défaut.
pick_list	[chaîne chaîne ... chaîne]	Permet à l'utilisateur d'importer la liste des nouvelles valeurs connues pour remplir la liste déroulante de la table. Voir l'exemple comme suit : set :reclassify.pick_list = [fruit dairy cereals]

Propriétés de reordernode



Le nœud Re-trier définit l'ordre naturel utilisé pour afficher les champs situés en aval. Cet ordre a une incidence sur l'affichage des champs en différents endroits : tableaux, listes et sélecteur de champs. Cette opération est utile lorsque vous utilisez des ensembles de données volumineux pour rendre plus visibles les champs intéressants. [Pour plus d'informations, reportez-vous à la section Nœud Re-trier dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create reordernode
set :reordernode.mode = Custom
set :reordernode.sort_by = Storage
set :reordernode.ascending = "false"
set :reordernode.start_fields = [Age Cholesterol]
```

```
set :reordernode.end_fields = [Drug]
```

propriétés reordernode	Le type de données	Description de la propriété
mode	Custom Auto	Vous pouvez trier les valeurs automatiquement ou indiquer un ordre personnalisé.
sort_by	Name Type Storage	
ascending	booléen	
start_fields	[champ1 champ2 ... champn]	Les nouveaux champs sont insérés après ces champs.
end_fields	[champ1 champ2 ... champn]	Les nouveaux champs sont insérés avant ces champs.

Propriétés de restructurenode



Le noeud Restructurer convertit un champ nominal ou un champ booléen en un groupe de champs renseignés à partir des valeurs d'un autre champ. Par exemple, si l'on considère un champ nommé *type de paiement*, qui comporte les valeurs *crédit*, *liquide* et *débit*, trois champs sont alors créés (*crédit*, *liquide*, *débit*), chacun contenant la valeur du paiement réel effectué. [Pour plus d'informations, reportez-vous à la section Noeud Restructurer dans le chapitre 4 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create restructurenode
connect :typenode to :restructurenode
set :restructurenode.fields_from.Drug = ["drugA" "drugX"]
set :restructurenode.include_field_name = "True"
set :restructurenode.value_mode = "OtherFields"
set :restructurenode.value_fields = ["Age" "BP"]
```

propriétés restructurenode	Le type de données	Description de la propriété
fields_from	[catégorie catégorie catégorie] all	Par exemple, set :restructurenode.fields_from.Drug = [drugA drugB] crée des champs appelés Drug_drugA et Drug_drugB. Pour utiliser toutes les catégories du champ spécifié : set :restructurenode.fields_from.Drug = all
include_field_name	booléen	Indique si le nom de champ doit être utilisé dans le nom de champ restructuré.
value_mode	OtherFields Flags	Indique le mode de spécification des valeurs pour les champs restructurés. Avec OtherFields, vous devez indiquer les champs à utiliser (voir ci-dessous). Avec Flags, les valeurs sont des booléens numériques.
value_fields	[champ champ champ]	Requis si value_mode est OtherFields. Indique les champs à utiliser en tant que champs de valeur.

Propriétés de rfmanalysisnode



Le nœud Analyse RFM (Récence, Effectif, Monétaire) permet de déterminer de façon quantitative les clients susceptibles d'être les meilleurs par l'étude de leur dernier achat (récence), l'effectif de leurs achats (effectif), et la somme dépensée lors de toutes les transactions (monétaire). [Pour plus d'informations, reportez-vous à la section Nœud Analyse RFM dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create rfmanalysisnode
connect :rfmaggregatenode to :rfmanalysisnode
set :rfmanalysisnode.recency = Recency
set :rfmanalysisnode.frequency = Frequency
set :rfmanalysisnode.monetary = Monetary
set :rfmanalysisnode.tied_values_method = Next
set :rfmanalysisnode.recalculate_bins = IfNecessary
set :rfmanalysisnode.recency_thresholds = [1, 500, 800, 1500, 2000, 2500]
```

propriétés rfmanalysisnode	Le type de données	Description de la propriété
recency	<i>champ</i>	Indiquez le champ de récence. Il peut s'agir d'une date, d'un horodatage ou d'un simple nombre.
frequency	<i>champ</i>	Indiquez le champ d'effectif.
monetary	<i>champ</i>	Spécifiez le champ monétaire.
recency_bins	<i>entier</i>	Indiquez le nombre d'intervalles de récence à générer.
recency_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données de récence. La valeur par défaut est 100.
frequency_bins	<i>entier</i>	Indiquez le nombre d'intervalles d'effectif à générer.
frequency_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données d'effectif. La valeur par défaut est 10.
monetary_bins	<i>entier</i>	Indiquez le nombre d'intervalles monétaires à générer.
monetary_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données monétaires. La valeur par défaut est 1.
tied_values_method	Next Current	Spécifiez l'intervalle dans lequel les données de valeur ex æquo doivent être placées.
recalculate_bins	Always IfNecessary	
add_outliers	<i>booléen</i>	Disponible uniquement si recalculate_bins est défini sur IfNecessary. Si cette valeur est définie, les enregistrements situés sous l'intervalle inférieur seront ajoutés à celui-ci et les enregistrements situés au-dessus de l'intervalle supérieur seront ajoutés à ce dernier.

propriétés rfanalysisnode	Le type de données	Description de la propriété
binned_field	Recency Frequency Monetary	
recency_thresholds	valeur valeur	Disponible uniquement si recalculate_bins est défini sur Always. Spécifiez les seuils supérieur et inférieur pour les intervalles de récence. Le seuil supérieur d'un intervalle est aussi le seuil inférieur de l'intervalle suivant ; par exemple, [10 30 60] définirait deux intervalles, le premier présentant des seuils supérieur et inférieur de 10 et 30, les seuils du deuxième intervalle étant de 30 et 60.
frequency_thresholds	valeur valeur	Disponible uniquement si recalculate_bins est défini sur Always.
monetary_thresholds	valeur valeur	Disponible uniquement si recalculate_bins est défini sur Always.

Propriétés de settoflagnode



Le noeud Binariser calcule plusieurs champs booléens en fonction des valeurs catégorielles définies pour un ou plusieurs champs nominaux. [Pour plus d'informations, reportez-vous à la section Noeud Binariser dans le chapitre 4 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create settoflagnode
connect :typenode to :settoflag
set :settoflagnode.fields_from Drug = ["drugA" "drugX"]
set :settoflagnode.true_value = "1"
set :settoflagnode.false_value = "0"
set :settoflagnode.use_extension = "True"
set :settoflagnode.extension = "Drug_Flag"
set :settoflagnode.add_as = Suffix
set :settoflagnode.aggregate = True
set :settoflagnode.keys = ['Cholesterol']
```

propriétés settoflagnode	Le type de données	Description de la propriété
fields_from	[<i>catégorie catégorie</i> <i>catégorie</i>] all	Par exemple, set :settoflagnode.fields_from Drug = [drugA drugB] crée des champs booléens appelés Drug_drugA et Drug_drugB. Pour utiliser toutes les catégories du champ spécifié : set :settoflagnode.fields_from Drug = all
true_value	chaîne	Indiquez la valeur true (vrai) utilisée par le noeud lors de la définition d'un booléen. La valeur par défaut est T.
false_value	chaîne	Indique la valeur false (faux) utilisée par le noeud lors de la définition d'un booléen. La valeur par défaut est F.

propriétés settoflagnode	Le type de données	Description de la propriété
use_extension	<i>booléen</i>	Utilisez une extension en tant que suffixe ou préfixe pour le nouveau champ booléen.
extension	<i>chaîne</i>	
add_as	Suffixe Préfixe	Indique si l'extension est ajoutée en tant que suffixe ou préfixe.
aggregate	<i>booléen</i>	Regroupe des enregistrements sur la base des champs-clés. Tous les champs booléens d'un groupe sont activés si un enregistrement est paramétré sur true (vrai).
keys	[<i>champ champ</i> <i>champ</i>]	Champs-clés.

Propriétés **statistictransformnode**



Le nœud Transformation exécute une sélection de commandes de syntaxe IBM® SPSS® Statistics en fonction des sources de données dans IBM® SPSS® Modeler. Ce nœud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Nœud Transformation Statistics dans le chapitre 8 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Les propriétés de ce nœud sont décrites dans [Propriétés statistictransformnode](#) sur p. 316.

Propriétés de **timeintervalsnode**



Le nœud Intervalle de temps définit des intervalles et crée, si nécessaire, des étiquettes pour la modélisation des séries temporelles. Si les valeurs ne sont pas espacées de manière égale, ce nœud peut les étoffer ou les agréger, selon les besoins, pour générer un intervalle uniforme entre les enregistrements. [Pour plus d'informations, reportez-vous à la section Nœud Intervalles de temps dans le chapitre 4 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create timeintervalsnode
set :timeintervalsnode.interval_type=SecondsPerDay
set :timeintervalsnode.days_per_week=4
set :timeintervalsnode.week_begins_on=Tuesday
set :timeintervalsnode.hours_per_day=10
set :timeintervalsnode.day_begins_hour=7
set :timeintervalsnode.day_begins_minute=5
set :timeintervalsnode.day_begins_second=17
set :timeintervalsnode.mode=Label
set :timeintervalsnode.year_start=2005
set :timeintervalsnode.month_start=January
set :timeintervalsnode.day_start=4
set :timeintervalsnode.pad.AGE=MeanOfRecentPoints
set :timeintervalsnode.agg_mode=Specify
```

```
set :timeintervalsnode.agg_set_default=Last
```

propriétés timeintervalsnode	Le type de données	Description de la propriété
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	Indique si vous souhaitez étiqueter les enregistrements de manière consécutive ou créer la série sur la base d'un champ de date, d'horodatage ou d'heure spécifié.
field	<i>champ</i>	Lorsque la série est créée à partir des données, indique le champ qui détermine la date ou l'heure de chaque enregistrement.
period_start	<i>entier</i>	Indique l'intervalle de début pour les périodes ou les périodes cycliques.
cycle_start	<i>entier</i>	Cycle de début pour les périodes cycliques.
year_start	<i>entier</i>	Pour les types d'intervalle concernés, année qui inclut le premier intervalle.
quarter_start	<i>entier</i>	Pour les types d'intervalle concernés, trimestre qui inclut le premier intervalle.
month_start	Janvier Février Mars Avril Mai Juin Juillet Août Septembre Octobre Novembre Décembre	
day_start	<i>entier</i>	
hour_start	<i>entier</i>	
minute_start	<i>entier</i>	
second_start	<i>entier</i>	
periods_per_cycle	<i>entier</i>	Pour les périodes cycliques, nombre de périodes au sein de chaque cycle.

propriétés timeintervalsnode	Le type de données	Description de la propriété
fiscal_year_begins	Janvier Février Mars Avril Mai Juin Juillet Août Septembre Octobre Novembre Décembre	Pour les intervalles trimestriels, indique le mois au cours duquel l'exercice commence.
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	Pour les intervalles périodiques (jours par semaine, heures par jour, minutes par jour et secondes par jour), indique le jour au cours duquel la semaine commence.
day_begins_hour	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique l'heure au cours de laquelle le jour commence. Il est possible d'utiliser cette propriété en combinaison avec day_begins_minute et day_begins_second pour indiquer une heure exacte telle que 8:05:01. Reportez-vous à l'exemple de syntaxe ci-dessous.
day_begins_minute	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique la minute au cours de laquelle le jour commence (par exemple, 5 dans 8:05).
day_begins_second	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique la seconde au cours de laquelle le jour commence (par exemple, 17 dans 8:05:17).
days_per_week	<i>entier</i>	Pour les intervalles périodiques (jours par semaine, heures par jour, minutes par jour et secondes par jour), indique le nombre de jours par semaine.
hours_per_day	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour et secondes par jour), indique le nombre d'heures par jour.

propriétés timeintervalsnode	Le type de données	Description de la propriété
interval_increment	1 2 3 4 5 6 10 15 20 30	Pour les minutes par jour et les secondes par jour, indique le nombre de minutes ou de secondes qui fait l'objet de l'incrément pour chaque enregistrement.
field_name_extension	chaîne	
field_name_extension_as_prefix	booléen	
date_format	"JJMMAA" "MMJJAA" "AAMMJJ" "AAAAMMJJ" AAAAJJJ DAY MONTH "JJ-MM-AA" "JJ-MM-AAAA" "MM-JJ-AA" "MM-JJ-AAAA" "JJ-MOI-AA" "JJ-MOI-AAAA" "AAAA-MM-JJ" "JJ.MM.AA" "JJ.MM.AAAA" "MM.JJ.AA" "MM.JJ.AAAA" "JJ.MOI.AA" "JJ.MOI.AAAA" "JJ/MM/AA" "JJ/MM/AAAA" "MM/JJ/AA" "MM/JJ/AAAA" "JJ/MOI/AA" "JJ/MOI/AAAA" MOIS AAAA t T AAAA ss SM AAAA	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	

propriétés timeintervalsnode	Le type de données	Description de la propriété
aggregate	Mean Sum Mode Min Max First Last TruelfAnyTrue	Indique la méthode d'agrégation pour un champ (par exemple, aggregate.AGE=Mean).
pad	Blank MeanOfRecentPoints True False	Indique la méthode de remplissage pour un champ (par exemple, pad.AGE=MeanOfRecentPoints).
agg_mode	All Specify	Indique si l'agrégation ou le remplissage de tous les champs doit avoir lieu à l'aide des fonctions par défaut (si nécessaire) ou si les champs et les fonctions à utiliser doivent être spécifiés.
agg_range_default	Mean Sum Mode Min Max	Indique la fonction par défaut à utiliser lors de l'agrégation de champs continus.
agg_set_default	Mode First Last	Indique la fonction par défaut à utiliser lors de l'agrégation de champs nominaux.
agg_flag_default	TruelfAnyTrue Mode First Last	
pad_range_default	Blank MeanOfRecentPoints	Indique la fonction par défaut à utiliser lors de l'extension de champs continus.
pad_set_default	Blank MostRecentValue	
pad_flag_default	Blank True False	
max_records_to_create	<i>entier</i>	Indique le nombre maximal d'enregistrements à créer lors du remplissage de la série.
estimation_from_beginning	<i>booléen</i>	
estimation_to_end	<i>booléen</i>	
estimation_start_offset	<i>entier</i>	
estimation_num_holdouts	<i>entier</i>	
create_future_records	<i>booléen</i>	
num_future_records	<i>entier</i>	
create_future_field	<i>booléen</i>	
future_field_name	<i>chaîne</i>	

Propriétés de transposenode



Le noeud Transposer fait passer les données des lignes vers les colonnes (et réciproquement) de sorte que les enregistrements deviennent des champs et les champs des enregistrements. [Pour plus d'informations, reportez-vous à la section Noeud Transposer dans le chapitre 4 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create transposenode
set :transposenode.transposed_names=Lire
set :transposenode.read_from_field="TimeLabel"
set :transposenode.max_num_fields="1000"
set :transposenode.id_field_name="ID"
```

propriétés transposenode	Le type de données	Description de la propriété
transposed_names	Prefix Read	Les nouveaux noms de champ peuvent être générés automatiquement à partir d'un préfixe défini ou ils peuvent être lus à partir d'un champ existant dans les données.
prefix	chaîne	
num_new_fields	entier	Lorsqu'un préfixe est utilisé, indique le nombre maximal de champs à créer.
read_from_field	champ	Champ à partir duquel les noms sont lus. Il doit s'agir d'un champ instancié ; sinon, une erreur apparaît lorsque le noeud est exécuté.
max_num_fields	entier	Lorsque des noms sont lus à partir d'un champ, indique une limite supérieure afin d'éviter de créer un nombre de champs trop important.
transpose_type	Numeric String Custom	Par défaut, seuls les champs continus (intervalle numérique) sont transposés, mais vous pouvez choisir un sous-ensemble personnalisé de champs numériques ou choisir de transposer tous les champs de type chaîne.
transpose_fields	[champ champ champ]	Indique les champs à transposer lorsque l'option Custom est utilisée.
id_field_name	champ	

Propriétés de typenode



Le noeud Typer définit les propriétés et métadonnées de champ. Par exemple, vous pouvez indiquer un niveau de mesure (continu, nominal, ordinal ou booléen) pour chaque champ, définir des options pour la gestion des valeurs manquantes et des valeurs système nulles, spécifier le rôle d'un champ en vue de la modélisation, définir des étiquettes de champ et de valeur, et indiquer les valeurs d'un champ. [Pour plus d'informations, reportez-vous à la section Noeud Typer dans le chapitre 4 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```

create typenode
connect :variablefilenode to :typenode
set :typenode.check.'Cholesterol' = Coerce
set :typenode.direction.'Drug' = Input
set :typenode.type.K = Range
set :typenode.values.Drug = [drugA drugB drugC drugD drugX drugY drugZ]
set :typenode.null_missing.BP = false
set :typenode.whitespace_missing.BP = "false"
set :typenode.description.BP = "Blood Pressure"
set :typenode.value_labels.BP = [{HIGH 'High Blood Pressure'}{NORMAL 'normal blood pressure'}]
set :typenode.display_places.K = 5
set :typenode.export_places.K = 2
set :typenode.grouping_symbol.Drug = None
set :typenode.column_width.Cholesterol = 25
set :typenode.justify.Cholesterol = Right

```

Dans certains cas, le nœud Typer doit être complètement instancié de façon à ce que d'autres nœuds fonctionnent correctement, notamment la propriété `fields from` du nœud Binariser. Vous pouvez simplement connecter un nœud Table et l'exécuter pour instancier les champs :

```

create tablenode
connect :typenode to :tablenode
execute :tablenode
delete :tablenode

```

propriétés typenode	Le type de données	Description de la propriété
direction	Input Target Both None Partition Split Frequency RecordID	Propriété saisie (rôles des champs). Syntaxe : NODE.direction.FIELDNAME <i>Remarque</i> : Les valeurs In et Out sont désormais remplacées. Leur prise en charge pourrait être supprimée dans une version ultérieure.
type	Range Flag Set Typeless Discrete Ordered Set Default	Type de champ. Définir type sur Default supprimera tous les paramètres values et si value_mode a la valeur Specify, il sera redéfini sur Read. Si value_mode est défini sur Pass ou Read, définir type n'aura pas d'effet sur value_mode. Syntaxe : NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	Propriété saisie en lecture seule pour le type de stockage de champ. Syntaxe : NODE.storage.FIELDNAME

propriétés typenode	Le type de données	Description de la propriété
check	None Nullify Coerce Discard Warn Abort	Propriété saisie (vérification du type et de l'intervalle des champs). Syntaxe : NODE.check.FIELDNAME
values	[<i>valeur valeur</i>]	Pour les champs continus , la première valeur représente la valeur minimale et la dernière valeur, la valeur maximale. Pour des champs nominaux, spécifiez toutes les valeurs. Pour les champs booléens, la première valeur représente <i>false</i> (faux) et la dernière valeur, <i>true</i> (vrai). La définition automatique de cette propriété paramètre la propriété <i>value_mode</i> sur <i>Specify</i> . Syntaxe : NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Détermine le mode de définition des valeurs. Remarque : vous ne pouvez pas paramétrer directement cette propriété sur <i>Specify</i> . Pour utiliser des valeurs spécifiques, paramétrez la propriété <i>values</i> . Syntaxe : NODE.value_mode.FIELDNAME
extend_values	<i>booléen</i>	S'applique quand <i>value_mode</i> est défini sur <i>Read</i> . Paramétrez cette valeur sur <i>T</i> pour ajouter des valeurs qui viennent d'être lues aux valeurs existantes du champ. Paramétrez cette valeur sur <i>F</i> pour supprimer des valeurs existantes en faveur des valeurs qui viennent d'être lues. Syntaxe : NODE.extend_values.FIELDNAME
enable_missing	<i>booléen</i>	Lorsque cette propriété est paramétrée sur <i>T</i> , elle active le suivi des valeurs manquantes du champ. Syntaxe : NODE.enable_missing.FIELDNAME
missing_values	[<i>valeur valeur ...</i>]	Spécifie les valeurs de données qui indiquent les données manquantes. Syntaxe : NODE.missing_values.FIELDNAME
range_missing	<i>booléen</i>	Indique si un intervalle de valeurs manquantes (non renseignées) a été défini pour un champ.
missing_lower	<i>chaîne</i>	Lorsque le paramètre <i>range_missing</i> a pour valeur <i>true</i> (vrai), indique la limite inférieure de l'intervalle des valeurs manquantes.
missing_upper	<i>chaîne</i>	Lorsque le paramètre <i>range_missing</i> a pour valeur <i>true</i> (vrai), indique la limite supérieure de l'intervalle des valeurs manquantes.

propriétés typenode	Le type de données	Description de la propriété
null_missing	booléen	Lorsque cette propriété est paramétrée sur T, les valeurs <i>null</i> (valeurs non définies affichées sous la forme \$null\$ dans le logiciel) sont considérées comme des valeurs manquantes. Syntaxe : NODE.null_missing.FIELDNAME
whitespace_missing	booléen	Lorsque cette propriété est paramétrée sur T, les valeurs composées uniquement de blancs (espaces, tabulations et caractères de nouvelle ligne) sont considérées comme des valeurs manquantes. Syntaxe : NODE.whitespace_missing.FIELDNAME
description	chaîne	Définit la description d'un champ.
value_labels	<i>[{Value LabelString} { Value LabelString} ...]</i>	Permet d'attribuer des étiquettes aux paires de valeurs. Voir l'exemple comme suit : set :typenode.value_labels.'Drug'=[{drugA label1} {drugB label2}]
display_places	entier	Définit le nombre de décimales du champ lorsqu'il est affiché (s'applique uniquement aux champs dont le stockage est REAL). La valeur -1 utilise le flux par défaut. Syntaxe : NODE.display_places.FIELDNAME
export_places	entier	Définit le nombre de décimales du champ lorsqu'il est exporté (s'applique uniquement aux champs dont le stockage est REAL). La valeur -1 utilise le flux par défaut. Syntaxe : NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD COMMA	Définit le séparateur décimal du champ (s'applique uniquement aux champs dont le stockage est REAL). Syntaxe : NODE.decimal_separator.FIELDNAME
date_format	"JJMAA" "MMJJAA" "AAMMJJ" "AAAAMMJJ" AAAAJJJ DAY MONTH "JJ-MM-AA" "JJ-MM-AAAA" "MM-JJ-AA" "MM-JJ-AAAA" "JJ-MOI-AA" "JJ-MOI-AAAA" "AAAA-MM-JJ" "JJ.MM.AA" "JJ.MM.AAAA" "MM.JJ.AA"	Définit le format de date du champ (s'applique uniquement aux champs dont le stockage est DATE ou TIMESTAMP). Syntaxe : NODE.date_format.FIELDNAME Voir l'exemple comme suit : set :tablenode.date_format.'LaunchDate' = "DDMMYY"

propriétés typenode	Le type de données	Description de la propriété
	"MM.JJ.AAAA" "JJ.MOI.AA" "JJ.MOI.AAAA" "JJ/MM/AA" "JJ/MM/AAAA" "MM/JJ/AA" "MM/JJ/AAAA" "JJ/MOI/AA" "JJ/MOI/AAAA" MOIS AAAA t T AAAA ss SM AAAA	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Définit le format horaire du champ (s'applique uniquement aux champs dont le stockage est TIME ou TIMESTAMP). Syntaxe : NODE.time_format.FIELDNAME Voir l'exemple comme suit : set :tablnode.time_format.'BOF_enter' = "HHMMSS"
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	Définit le format d'affichage des nombres du champ. Syntaxe : NODE.number_format.FIELDNAME
standard_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format standard. La valeur -1 utilise le flux par défaut. La propriété display_places existante modifie également ce nombre, mais a été remplacée. Syntaxe : NODE.standard_places.FIELDNAME
scientific_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format scientifique. La valeur -1 utilise le flux par défaut. Syntaxe : NODE.scientific_places.FIELDNAME
currency_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format monétaire. La valeur -1 utilise le flux par défaut. Syntaxe : NODE.currency_places.FIELDNAME
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	Définit le symbole de regroupement du champ. Syntaxe : NODE.grouping_symbol.FIELDNAME

propriétés typenode	Le type de données	Description de la propriété
column_width	<i>entier</i>	Définit la largeur des colonnes du champ. La valeur -1 définit la largeur des colonnes sur Auto. Syntaxe : NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	Définit la justification des colonnes du champ. Syntaxe : NODE.justify.FIELDNAME

Propriétés des noeuds Graphiques

Propriétés communes aux noeuds Graphiques

Cette section décrit les propriétés disponibles pour les noeuds Graphiques, notamment les propriétés communes, ainsi que celles propres à chaque type de noeud.

Propriétés communes aux noeuds Graphiques	Le type de données	Description de la propriété
title	chaîne	Spécifie le titre. Exemple : « Voici un titre. »
caption	chaîne	Spécifie la légende. Exemple : « Voici une légende. »
output_mode	Screen File	Indique si la sortie d'un noeud Graphiques doit être affichée ou écrite dans un fichier.
output_format	BMP JPEG PNG HTML output (.cou)	Indique le type de sortie. Le type exact de sortie autorisée varie pour chaque noeud.
full_filename	chaîne	Indique le chemin et le nom de fichier cible de la sortie générée à partir du noeud Graphiques.
use_graph_size	booléen	Vérifie si le graphique est dimensionné de manière explicite, à l'aide des propriétés de largeur et de hauteur définies ci-dessous. Ne concerne que les graphiques affichés à l'écran. Non disponible pour le noeud Proportion.
graph_width	nombre	Lorsque use_graph_size prend la valeur True, définit la largeur du graphique en pixels.
graph_height	nombre	Lorsque use_graph_size prend la valeur True, définit la hauteur du graphique en pixels.

Remarques

Désactivation des champs facultatifs. Vous pouvez désactiver les champs facultatifs (par exemple, un champ de superposition pour les nuages) en paramétrant la valeur de la propriété sur "" (chaîne vide), comme le montre l'exemple suivant.

```
set :plotnode.color_field = ""
```

Indication des couleurs. Vous pouvez indiquer les couleurs des titres, légendes, arrière-plans et étiquettes à l'aide des chaînes hexadécimales commençant par le symbole dièse (#). Par exemple, pour mettre l'arrière-plan du graphique en bleu ciel, utilisez l'instruction suivante :

```
set mygraph.graph_background="#87CEEB"
```

Les deux premiers caractères, 87, indiquent le contenu rouge, les deux caractères du milieu, CE, indiquent le contenu vert, et les deux derniers caractères, EB, indiquent le contenu bleu. Chaque caractère peut correspondre à une valeur comprise entre 0 et 9 ou A et F. Si elles sont associées, ces valeurs peuvent indiquer une couleur RVB (rouge, vert, bleu).

Remarque : Lorsque vous spécifiez des couleurs en RVB, vous pouvez utiliser le sélecteur de champs de l'interface utilisateur pour déterminer le code couleur correct. Déplacez la souris au-dessus de la couleur pour activer l'info-bulle contenant les informations souhaitées.

Propriétés de collectionnode



Le noeud Résumé fournit la proportion de valeurs d'un champ numérique par rapport aux valeurs d'un autre champ. (Il génère des graphiques semblables aux histogrammes.) Il est utile pour illustrer une variable ou un champ dont les valeurs changent avec le temps. Grâce à la représentation graphique en 3D, vous pouvez en outre inclure un axe symbolique affichant les proportions par catégorie. [Pour plus d'informations, reportez-vous à la section Onglet nuage de Résumé dans le chapitre 5 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create collectionnode
position :collectionnode at ^posX ^posY
#"Plot" tab
set :collectionnode.three_D = True
set :collectionnode.collect_field = 'Drug'
set :collectionnode.over_field = 'Age'
set :collectionnode.by_field = 'BP'
set :collectionnode.operation = Sum
#"Overlay" section
set :collectionnode.color_field = 'Drug'
set :collectionnode.panel_field = 'Sex'
set :collectionnode.animation_field = ''
#"Options" tab
set :collectionnode.range_mode = Automatic
set :collectionnode.range_min = 1
set :collectionnode.range_max = 100
set :collectionnode.bins = ByNumber
set :collectionnode.num_bins = 10
set :collectionnode.bin_width = 5
```

propriétés collectionnode	Le type de données	Description de la propriété
over_field	champ	
over_label_auto	booléen	
over_label	chaîne	
collect_field	champ	
collect_label_auto	booléen	
collect_label	chaîne	
three_D	booléen	
by_field	champ	

propriétés collectionnode	Le type de données	Description de la propriété
by_label_auto	booléen	
by_label	chaîne	
operation	Sum Mean Min Max SDev	
color_field	chaîne	
panel_field	chaîne	
animation_field	chaîne	
range_mode	Automatic UserDefined	
range_min	nombre	
range_max	nombre	
bins	ByNumber ByWidth	
num_bins	nombre	
bin_width	nombre	
use_grid	booléen	
graph_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.
page_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.

Propriétés de distributionnode



Le noeud Proportion fournit l'occurrence des valeurs symboliques (catégorielles), comme un type de prêt hypothécaire ou le sexe d'un individu. Ce noeud est souvent utilisé pour montrer les déséquilibres des données, déséquilibres que vous pouvez rectifier à l'aide d'un noeud Equilibrer avant la création d'un modèle. [Pour plus d'informations, reportez-vous à la section Noeud Proportion dans le chapitre 5 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create distributionnode
# "Plot" tab
set :distributionnode.plot = Flags
set :distributionnode.x_field = 'Age'
set :distributionnode.color_field = 'Drug'
set :distributionnode.normalize = True
set :distributionnode.sort_mode = ByOccurrence
set :distributionnode.use_proportional_scale = True
```

propriétés distributionnode	Le type de données	Description de la propriété
plot	SelectedFields Flags	
x_field	champ	

propriétés distributionnode	Le type de données	Description de la propriété
color_field	champ	Champ de superposition.
normalize	booléen	
sort_mode	ByOccurrence Alphabetic	
use_proportional_scale	booléen	

Propriétés de evaluationnode



Le noeud Evaluation permet d'évaluer et de comparer des modèles prédictifs. Le graphique d'évaluation montre l'aptitude des modèles à prédire des résultats spécifiques. Il trie les enregistrements en fonction de la valeur prédite et de la confiance dans cette prévision. Il scinde les enregistrements en groupes de taille égale (**quantiles**), puis reporte la valeur du critère traité pour chaque quantile, du plus élevé au plus faible. Les divers modèles apparaissent sous forme de lignes dans le graphique. [Pour plus d'informations, reportez-vous à la section Noeud Evaluation dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create evaluationnode
position :evaluationnode at ^posX ^posY
# "Plot" tab
set :evaluationnode.chart_type = Gains
set :evaluationnode.cumulative = False
set :evaluationnode.field_detection_method = Name
set :evaluationnode.inc_baseline = True
set :evaluationnode.n_tile = Deciles
set :evaluationnode.style = Point
set :evaluationnode.point_type = Dot
set :evaluationnode.use_fixed_cost = True
set :evaluationnode.cost_value = 5.0
set :evaluationnode.cost_field = 'Na'
set :evaluationnode.use_fixed_revenue = True
set :evaluationnode.revenue_value = 30.0
set :evaluationnode.revenue_field = 'Age'
set :evaluationnode.use_fixed_weight = True
set :evaluationnode.weight_value = 2.0
set :evaluationnode.weight_field = 'K'
```

propriétés evaluationnode	Le type de données	Description de la propriété
chart_type	Gains Response Lift Profit ROI	
inc_baseline	booléen	
field_detection_method	Metadata Name	
use_fixed_cost	booléen	

propriétés evaluationnode	Le type de données	Description de la propriété
cost_value	<i>nombre</i>	
cost_field	<i>chaîne</i>	
use_fixed_revenue	<i>booléen</i>	
revenue_value	<i>nombre</i>	
revenue_field	<i>chaîne</i>	
use_fixed_weight	<i>booléen</i>	
weight_value	<i>nombre</i>	
weight_field	<i>champ</i>	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	<i>booléen</i>	
style	Line Point	
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Usine Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
export_data	<i>booléen</i>	
data_filename	<i>chaîne</i>	
delimiter	<i>chaîne</i>	
new_line	<i>booléen</i>	
inc_field_names	<i>booléen</i>	
inc_best_line	<i>booléen</i>	
inc_business_rule	<i>booléen</i>	
business_rule_condition	<i>chaîne</i>	
plot_score_fields	<i>booléen</i>	
score_fields	<i>[champ1 ... champN]</i>	
target_field	<i>champ</i>	
use_hit_condition	<i>booléen</i>	

propriétés evaluationnode	Le type de données	Description de la propriété
hit_condition	chaîne	
use_score_expression	booléen	
score_expression	chaîne	
caption_auto	booléen	

Propriétés de graphboardnode



Le noeud Représentation graphique propose différents types de graphiques dans un noeud unique. Ce noeud permet de choisir les champs de données que vous souhaitez explorer puis de sélectionner un graphique parmi ceux disponibles pour les données sélectionnées. Le noeud filtre automatiquement tous les types de graphiques ne fonctionnant pas avec les sélections de champs. [Pour plus d'informations, reportez-vous à la section Noeud Représentation Graphique dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Remarque : Si vous définissez une propriété non valide pour le type de graphique (par exemple, si vous spécifiez y_field pour un histogramme), cette propriété est ignorée.

Exemple

```
create graphboardnode
connect DRUG4n to :graphboardnode
set :graphboardnode.graph_type="Line"
set :graphboardnode.x_field = "K"
set :graphboardnode.y_field = "Na"
execute :graphboardnode
```

propriétés graphboard	Le type de données	Description de la propriété
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians	Identifie le type de graphique.

propriétés graphboard	Le type de données	Description de la propriété
	CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap PointsOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	
x_field	champ	Spécifie une étiquette personnalisée pour l'axe x. Disponible uniquement pour les étiquettes.
y_field	champ	Spécifie une étiquette personnalisée pour l'axe y. Disponible uniquement pour les étiquettes.
z_field	champ	Utilisé dans certains graphiques en 3D
color_field	champ	Utilisé dans les cartes de zones de chaleurs.
size_field	champ	Utilisé dans les graphiques à bulles.
categories_field	champ	
values_field	champ	
rows_field	champ	
columns_field	champ	
fields	champ	
start_longitude	champ	Utilisé avec les flèches sur une carte de référence.
end_longitude	champ	
start_latitude	champ	
end_latitude	champ	
data_key_field	champ	Utilisé dans diverses cartes.
panelrow_field	chaîne	
panelcol_field	chaîne	
animation_field	chaîne	
longitude	champ	Utilisé avec les coordonnées sur les cartes.
latitude	champ	
map_color_field	champ	

Propriétés de histogramnode



Le noeud Histogramme montre l'occurrence des valeurs des champs numériques. Il est souvent utilisé pour explorer les données avant toute création de modèles ou manipulation. Semblable au noeud Proportion, le noeud Histogramme sert souvent à montrer les déséquilibres des données. [Pour plus d'informations, reportez-vous à la section Onglet Nuage d'histogramme dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create histogramnode
position :histogramnode at ^posX ^posY
#"Plot" tab
set :histogramnode.field = 'Drug'
set :histogramnode.color_field = 'Drug'
set :histogramnode.panel_field = 'Sex'
set :histogramnode.animation_field = ''
#"Options" tab
set :histogramnode.range_mode = Automatic
set :histogramnode.range_min = 1.0
set :histogramnode.range_max = 100.0
set :histogramnode.num_bins = 10
set :histogramnode.bin_width = 10
set :histogramnode.normalize = True
set :histogramnode.separate_bands = False
```

propriétés histogramnode	Le type de données	Description de la propriété
field	champ	
color_field	champ	
panel_field	champ	
animation_field	champ	
range_mode	Automatic UserDefined	
range_min	nombre	
range_max	nombre	
bins	ByNumber ByWidth	
num_bins	nombre	
bin_width	nombre	
normalize	booléen	
separate_bands	booléen	
x_label_auto	booléen	
x_label	chaîne	
y_label_auto	booléen	
y_label	chaîne	
use_grid	booléen	
graph_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.

propriétés histogramnode	Le type de données	Description de la propriété
page_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.
normal_curve	booléen	Indique si la courbe de proportion normale doit apparaître sur la sortie.

Propriétés de multiplotnode



Le noeud Courbes génère un graphique qui affiche plusieurs champs *Y* pour un seul champ *X*. Les champs *Y* sont représentés par des lignes colorées. Chacun équivaut à un noeud Nuage dont le style est défini sur Ligne et le mode *X* sur Trier. Les graphiques Courbes sont utiles lorsque vous souhaitez étudier la fluctuation de plusieurs variables au fil du temps. [Pour plus d'informations, reportez-vous à la section Noeud Courbes dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create multiplotnode
# "Plot" tab
set :multiplotnode.x_field = 'Age'
set :multiplotnode.y_fields = ['Drug' 'BP']
set :multiplotnode.panel_field = 'Sex'
# "Overlay" section
set :multiplotnode.animation_field = ''
set :multiplotnode.tooltip = "test"
set :multiplotnode.normalize = True
set :multiplotnode.use_overlay_expr = False
set :multiplotnode.overlay_expression = "test"
set :multiplotnode.records_limit = 500
set :multiplotnode.if_over_limit = PlotSample
```

propriétés multiplotnode	Le type de données	Description de la propriété
x_field	champ	
y_fields	[champ champ champ]	
panel_field	champ	
animation_field	champ	
normalize	booléen	
use_overlay_expr	booléen	
overlay_expression	chaîne	
records_limit	nombre	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	booléen	
x_label	chaîne	
y_label_auto	booléen	
y_label	chaîne	

propriétés multiplotnode	Le type de données	Description de la propriété
use_grid	<i>booléen</i>	
graph_background	<i>Couleur</i>	Les couleurs de graphique standard sont décrites au début de cette section.
page_background	<i>Couleur</i>	Les couleurs de graphique standard sont décrites au début de cette section.

Propriétés de plotnode



Le noeud Nuage montre les relations existant entre les champs numériques. Vous pouvez créer un graphique Nuage à l'aide de points (diagramme de dispersion) ou de lignes. [Pour plus d'informations, reportez-vous à la section Noeud Nuage dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create plotnode
# "Plot" tab
set :plotnode.three_D = True
set :plotnode.x_field = 'BP'
set :plotnode.y_field = 'Cholesterol'
set :plotnode.z_field = 'Drug'
# "Overlay" section
set :plotnode.color_field = 'Drug'
set :plotnode.size_field = 'Age'
set :plotnode.shape_field = ''
set :plotnode.panel_field = 'Sex'
set :plotnode.animation_field = 'BP'
set :plotnode.transp_field = ''
set :plotnode.style = Point
# "Output" tab
set :plotnode.output_mode =
set :plotnode.output_format = JPEG
set :plotnode.full_filename = "C:/Temp/Graph_Output/plot_output.jpeg"
```

propriétés plotnode	Le type de données	Description de la propriété
x_field	<i>champ</i>	Spécifie une étiquette personnalisée pour l'axe <i>x</i> . Disponible uniquement pour les étiquettes.
y_field	<i>champ</i>	Spécifie une étiquette personnalisée pour l'axe <i>y</i> . Disponible uniquement pour les étiquettes.
three_D	<i>booléen</i>	Spécifie une étiquette personnalisée pour l'axe <i>y</i> . Disponible uniquement pour les étiquettes des graphiques en 3D.
z_field	<i>champ</i>	
color_field	<i>champ</i>	Champ de superposition.
size_field	<i>champ</i>	
shape_field	<i>champ</i>	

propriétés plotnode	Le type de données	Description de la propriété
panel_field	<i>champ</i>	Sélectionne un champ nominal ou un champ booléen pour créer un diagramme distinct pour chaque catégorie. Les diagrammes apparaissent tous sous forme de panneaux dans une fenêtre de sortie.
animation_field	<i>champ</i>	Spécifie un champ nominal ou un champ booléen pour mettre en évidence les catégories des valeurs de données en créant une série de diagrammes animés apparaissant les uns après les autres.
transp_field	<i>champ</i>	Désigne un champ destiné à mettre en évidence les catégories des valeurs de données en utilisant un degré de transparence différent pour chaque catégorie. Non disponible pour les nuages de lignes.
overlay_type	None Smoother Function	Indique si une fonction de superposition ou un lissage LOESS est affiché.
overlay_expression	<i>chaîne</i>	Indique l'expression utilisée lorsque <i>overlay_type</i> est paramétré sur <i>Function</i> .
style	Point Line	
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Usine Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	<i>nombre</i>	
x_range_max	<i>nombre</i>	
y_range_mode	Automatic UserDefined	
y_range_min	<i>nombre</i>	
y_range_max	<i>nombre</i>	

propriétés plotnode	Le type de données	Description de la propriété
z_range_mode	Automatic UserDefined	
z_range_min	nombre	
z_range_max	nombre	
jitter	booléen	
records_limit	nombre	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	booléen	
x_label	chaîne	
y_label_auto	booléen	
y_label	chaîne	
z_label_auto	booléen	
z_label	chaîne	
use_grid	booléen	
graph_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.
page_background	Couleur	Les couleurs de graphique standard sont décrites au début de cette section.
use_overlay_expr	booléen	Désormais obsolète et remplacé par overlay_type.

Propriétés de timeplotnode



Le noeud Tracé horaire affiche un ou plusieurs ensembles de données temporelles. En règle générale, vous utilisez un noeud Intervalles de temps, en premier lieu, pour créer un champ *TimeLabel* qui servira d'étiquette à l'axe x. [Pour plus d'informations, reportez-vous à la section Noeud Tracé horaire dans le chapitre 5 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create timeplotnode
set :timeplotnode.y_fields = ['sales' 'men' 'women']
set :timeplotnode.panel = True
set :timeplotnode.normalize = True
set :timeplotnode.line = True
set :timeplotnode.smoother = True
set :timeplotnode.use_records_limit = True
set :timeplotnode.records_limit = 2000
# Appearance settings
set :timeplotnode.symbol_size = 2.0
```

propriétés timeplotnode	Le type de données	Description de la propriété
plot_series	Series Models	
use_custom_x_field	booléen	
x_field	champ	
y_fields	[champ champ champ]	
panel	booléen	
normalize	booléen	
line	booléen	
points	booléen	
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Usine Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
smoother	booléen	Pour pouvoir ajouter des fonctions de lissage au graphique, paramétrez panel sur True (vrai).
use_records_limit	booléen	
records_limit	entier	
symbol_size	nombre	Définit une taille de symbole. Par exemple, set :webnode.symbol_size = 5.5 permet de créer un symbole dont la taille est supérieure à la taille par défaut.
panel_layout	Horizontal Vertical	

Propriétés de webnode



Le noeud Relations illustre la force de la relation existant entre les valeurs de plusieurs champs symboliques (catégoriels). Le graphique utilise des lignes d'épaisseur différente pour représenter les forces de connexion. Par exemple, vous pouvez utiliser un noeud Relations pour explorer la relation avec l'achat d'un ensemble d'articles sur un site de commerce électronique. [Pour plus d'informations, reportez-vous à la section Noeud Relations dans le chapitre 5 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create webnode
# "Plot" tab
set :webnode.use_directed_web = True
set :webnode.to_field = 'Drug'
set :webnode.fields = ['BP' 'Cholesterol' 'Sex' 'Drug']
set :webnode.from_fields = ['BP' 'Cholesterol' 'Sex']
set :webnode.true_flags_only = False
set :webnode.line_values = Absolute
set :webnode.strong_links_heavier = True
# "Options" tab
set :webnode.max_num_links = 300
set :webnode.links_above = 10
set :webnode.num_links = ShowAll
set :webnode.discard_links_min = True
set :webnode.links_min_records = 5
set :webnode.discard_links_max = True
set :webnode.weak_below = 10
set :webnode.strong_above = 19
set :webnode.link_size_continuous = True
set :webnode.web_display = Circular
```

propriétés webnode	Le type de données	Description de la propriété
use_directed_web	booléen	
fields	[champ champ champ]	
to_field	champ	
from_fields	[champ champ champ]	
true_flags_only	booléen	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	booléen	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	nombre	
links_above	nombre	

propriétés webnode	Le type de données	Description de la propriété
discard_links_min	<i>booléen</i>	
links_min_records	<i>nombre</i>	
discard_links_max	<i>booléen</i>	
links_max_records	<i>nombre</i>	
weak_below	<i>nombre</i>	
strong_above	<i>nombre</i>	
link_size_continuous	<i>booléen</i>	
web_display	Circular Network Directed Grid	
graph_background	<i>Couleur</i>	Les couleurs de graphique standard sont décrites au début de cette section.
symbol_size	<i>nombre</i>	Définit une taille de symbole. Par exemple, set :webnode.symbol_size = 5.5 permet de créer un symbole dont la taille est supérieure à la taille par défaut.

Propriétés des noeuds de modélisation

Propriétés communes des noeuds de modélisation

Les propriétés suivantes sont communes à certains ou à tous les noeuds de modélisation. Les éventuelles exceptions sont notées dans la documentation des noeuds de modélisation individuels, le cas échéant.

Propriété	Valeurs	Description de la propriété
custom_fields	<i>booléen</i>	Si la valeur est définie sur true (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le nœud actuel. Si elle est définie sur false (faux), les paramètres actuels provenant d'un nœud Typer en amont sont utilisés.
target ou targets	<i>champ</i> ou <i>[champ1 ... champN]</i>	Spécifie un ou plusieurs champs cibles selon le type de modèle.
inputs	<i>[champ1 ... champN]</i>	Champs d'entrée ou variable indépendante utilisés par le modèle.
partition	<i>champ</i>	
use_partitioned_data	<i>booléen</i>	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
use_split_data	<i>booléen</i>	
splits	<i>[champ1 ... champN]</i>	Indique le champ ou les champs à utiliser pour la modélisation découpée. Efficace uniquement si use_split_data est défini sur True.
use_frequency	<i>booléen</i>	Les champs de pondération et d'effectif sont utilisés par des modèles spécifiques comme indiqué pour chaque type de modèle.
frequency_field	<i>champ</i>	
use_weight	<i>booléen</i>	
weight_field	<i>champ</i>	
use_model_name	<i>booléen</i>	
model_name	<i>chaîne</i>	Nom personnalisé du nouveau modèle.
mode	Simple Expert	

Propriétés de anomalydetectionnode



Le noeud Détection des anomalies identifie les observations inhabituelles, ou valeurs éloignées, qui ne se conforment pas aux patrons de données « normales ». Il vous permet d'identifier les valeurs éloignées même si celles-ci ne correspondent pas aux patrons connus précédemment et même si vous ne savez pas exactement ce que vous recherchez. [Pour plus d'informations, reportez-vous à la section Noeud Détection des anomalies dans le chapitre 4 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create anomalydetectionnode
set :anomalydetectionnode.anomaly_method=PerRecords
set :anomalydetectionnode.percent_records=95
set :anomalydetectionnode.mode=Expert
set :anomalydetectionnode.peer_group_num_auto=true
set :anomalydetectionnode.min_num_peer_groups=3
set :anomalydetectionnode.max_num_peer_groups=10
```

Propriétés anomalydetectionnode	Valeurs	Description de la propriété
inputs	[<i>champ1 ... champN</i>]	Les modèles Détection des anomalies filtrent les enregistrements en fonction des champs d'entrée spécifiés. Ils n'utilisent pas de champ cible. Les champs de pondération et d'effectifs également ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	Indique la méthode utilisée pour déterminer la valeur de césure de signalement des enregistrements jugés comme irréguliers.
index_level	<i>nombre</i>	Indique la valeur de césure pour le signalement des anomalies.
percent_records	<i>nombre</i>	Définit le seuil de signalement des enregistrements, basé sur le pourcentage d'enregistrements dans les données d'apprentissage.
num_records	<i>nombre</i>	Définit le seuil de signalement des enregistrements, basé sur le nombre d'enregistrements dans les données d'apprentissage.
num_fields	<i>entier</i>	Nombre de champs à signaler pour chaque enregistrement irrégulier.
impute_missing_values	<i>booléen</i>	
adjustment_coeff	<i>nombre</i>	Valeur utilisée pour équilibrer la pondération relative attribuée aux champs continus et aux champs catégoriels pour les calculs de distance.

Propriétés anomalydetectionnode	Valeurs	Description de la propriété
peer_group_num_auto	booléen	Calcule automatiquement le nombre de groupes d'homologues.
min_num_peer_groups	entier	Indique le nombre minimal de groupes d'homologues utilisés quand la propriété peer_group_num_auto est définie sur True (vrai).
max_num_per_groups	entier	Indique le nombre maximal de groupes d'homologues.
num_peer_groups	entier	Indique le nombre de groupes d'homologues utilisés quand la propriété peer_group_num_auto est définie sur False (faux).
noise_level	nombre	Détermine le mode de traitement des valeurs éloignées au cours de la classification non supervisée. Spécifiez une valeur comprise entre 0 et 0.5.
noise_ratio	nombre	Indique la quantité de mémoire allouée au composant qui doit être utilisée pour la mise en mémoire tampon du bruit. Spécifiez une valeur comprise entre 0 et 0.5.

Propriétés de apriorinode



Le noeud Apriori extrait des données un ensemble de règles et retient les règles contenant la plus grande quantité d'informations. Le noeud Apriori fournit cinq méthodes de sélection de règles et utilise un modèle d'indexation sophistiqué pour traiter efficacement les volumes de données importants. Pour les problèmes importants, l'apprentissage du noeud Apriori est généralement plus rapide ; il n'existe aucune limite quant au nombre de règles pouvant être conservées et il peut prendre en charge des règles faisant l'objet de 32 pré-conditions. Le noeud Apriori exige que les champs d'entrée et de sortie soient tous catégoriels, mais fournit de meilleures performances car il est optimisé de ce type de données. [Pour plus d'informations, reportez-vous à la section Noeud Apriori dans le chapitre 12 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create apriorinode
# "Fields" tab
set :apriorinode.custom_fields = True
set :apriorinode.use_transactional_data = True
set :apriorinode.id_field = 'Age'
set :apriorinode.contiguous = True
set :apriorinode.content_field = 'Drug'
# Ceux-ci semblent avoir été modifiés, ils sont normalement :
#help set :apriorinode.consequents = ['Age']
#help set :apriorinode.antecedents = ['BP' 'Cholesterol' 'Drug']
# now it seems we have;
#help set :apriorinode.content = ['Age']
set :apriorinode.partition = Test
```

```

# "Model" tab
set :apriorinode.use_model_name = False
set :apriorinode.model_name = "Apriori_bp_choles_drug"
set :apriorinode.min_supp = 7.0
set :apriorinode.min_conf = 30.0
set :apriorinode.max_antecedents = 7
set :apriorinode.true_flags = False
set :apriorinode.optimize = Memory
# "Expert" tab
set :apriorinode.mode = Expert
set :apriorinode.evaluation = ConfidenceRatio
set :apriorinode.lower_bound = 7

```

Propriétés apriorinode	Valeurs	Description de la propriété
consequents	<i>champ</i>	Les modèles Apriori utilisent les Conséquences et les Antécédents à la place des champs cible et entrée standard. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
antecedents	[<i>champ1 ... champN</i>]	
min_supp	<i>nombre</i>	
min_conf	<i>nombre</i>	
max_antecedents	<i>nombre</i>	
true_flags	<i>booléen</i>	
optimize	Speed Memory	
use_transactional_data	<i>booléen</i>	
contiguous	<i>booléen</i>	
id_field	<i>chaîne</i>	
content_field	<i>chaîne</i>	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	<i>nombre</i>	
optimize	Speed Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.

Propriétés autotclassifiernode



Le noeud Classificateur automatique crée et compare les résultats binaires de plusieurs modèles différents (oui ou non, avec ou sans attrition, etc.), ce qui vous permet de choisir la meilleure approche pour une analyse donnée. Plusieurs algorithmes de modélisation sont pris en charge. Vous pouvez alors sélectionner les méthodes que vous souhaitez utiliser, les options spécifiques pour chacune d'elles et le critère de comparaison des résultats. Le noeud génère un ensemble de modèles basé sur les options spécifiées et classe les meilleurs candidats en fonction des critères indiqués.

Pour plus d'informations, reportez-vous à la section Noeud Classificateur automatique dans le chapitre 5 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.

Exemple

```
create autotclassifiernode
set :autotclassifiernode.ranking_measure=Accuracy
set :autotclassifiernode.ranking_dataset=Training
set :autotclassifiernode.enable_accuracy_limit=true
set :autotclassifiernode.accuracy_limit=0.9
set :autotclassifiernode.calculate_variable_importance=true
set :autotclassifiernode.use_costs=true
set :autotclassifiernode.svm=false
```

Propriétés autotclassifiernode	Valeurs	Description de la propriété
target	champ	Pour les cibles de type booléen, le noeud Classificateur automatique requiert un seul champ cible et un ou plusieurs champs d'entrée. Il est également possible de spécifier des champs de pondération et d'effectifs. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	entier	Nombre de modèles à inclure dans le nugget de modèle. Indiquez un entier compris entre 1 et 100.
calculate_variable_importance	booléen	
enable_accuracy_limit	booléen	
accuracy_limit	entier	Entier compris entre 0 et 100.
enable_area_under_curve_limit	booléen	
area_under_curve_limit	nombre	Nombre réel compris entre 0,0 et 1,0.
enable_profit_limit	booléen	
profit_limit	nombre	Entier supérieur à 0.
enable_lift_limit	booléen	
lift_limit	nombre	Nombre réel supérieur à 1.0.

Propriétés autotransformateur	Valeurs	Description de la propriété
enable_number_of_variables_limit	booléen	
number_of_variables_limit	nombre	Entier supérieur à 0.
use_fixed_cost	booléen	
fixed_cost	nombre	Nombre réel supérieur à 0.0.
variable_cost	champ	
use_fixed_revenue	booléen	
fixed_revenue	nombre	Nombre réel supérieur à 0.0.
variable_revenue	champ	
use_fixed_weight	booléen	
fixed_weight	nombre	Nombre réel supérieur à 0,0.
variable_weight	champ	
lift_percentile	nombre	Entier compris entre 0 et 100.
enable_model_build_time_limit	booléen	
model_build_time_limit	nombre	Entier défini sur le nombre de minutes devant être utilisé pour limiter le temps nécessaire à la création de chaque modèle.
enable_stop_after_time_limit	booléen	
stop_after_time_limit	nombre	Nombre réel défini sur le nombre d'heures servant à limiter le temps global passé pour une exécution de classificateur automatique.
enable_stop_after_valid_model_produced	booléen	
use_costs	booléen	
<algorithm>	booléen	Active ou désactive l'utilisation d'un algorithme particulier, par exemple : set :autotransformateur.chaid=true
<algorithm>.<property>	chaîne	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, reportez-vous à la section Définition des propriétés de l'algorithme sur p. 202.

Définition des propriétés de l'algorithme

Pour les noeuds Classificateur automatique, Numérisation automatique et Classification automatique, les propriétés d'algorithmes spécifiques utilisées par le noeud peuvent être définies à l'aide de la forme générale :

```
set :autotransformateur.<algorithm>.<property> = <value>
```

```
set :autonumericnode.<algorithm>.<property> = <value>
```

```
set :autoclusternode.<algorithm>.<property> = <value>
```


Par exemple :

```
set :autoclassifiernode.neuralnetwork.method = MultilayerPerceptron
```

Les noms des algorithmes pour le noeud Classificateur automatique sont `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` et `knn`.

Les noms des algorithmes pour le noeud Classificateur numérique sont `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` et `knn`.

Les noms des algorithmes pour le noeud Classification automatique sont `twostep`, `k-means` et `kohonen`.

Les noms de propriété sont standard (reportez-vous aux informations relatives à chaque noeud d'algorithme).

Les propriétés d'algorithme qui contiennent des points ou d'autres signes de ponctuation doivent être placés entre guillemets simples (' '), par exemple :

```
set :autoclassifiernode.logreg.tolerance = '1.0E-5'
```

Plusieurs valeurs peuvent également être affectées à une propriété, par exemple :

```
set :autoclassifiernode.decisionlist.search_direction = [Up Down]
```

Pour activer ou désactiver l'utilisation d'un algorithme particulier :

```
set :autoclassifiernode.chaid=true
```

Remarques :

- Utilisez des minuscules lors de la définition des valeurs `true` (vrai) et `false` (faux) (plutôt que `False`).
- Lorsque certaines options d'algorithme ne sont pas disponibles dans le noeud Classificateur automatique ou qu'une seule valeur peut être indiquée plutôt qu'un intervalle de valeurs, les mêmes limites s'appliquent à la génération de scripts que lorsque l'on accède au noeud de la manière standard.

Propriétés du noeud de classification automatique



Le noeud Classification automatique évalue et compare les modèles de classification identifiant des groupes d'enregistrements ayant des caractéristiques similaires. Le noeud fonctionne de la même manière que les autres noeuds de modélisation automatiques, vous permettant de tester plusieurs combinaisons d'options en une seule modélisation. Les modèles peuvent être comparés à l'aide de mesures de bases permettant d'essayer de filtrer et de classer l'utilité des modèles de classification et de fournir une mesure en fonction de l'importance de champs particuliers. [Pour plus d'informations, reportez-vous à la section Noeud Classification automatique dans le chapitre 5 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```

create autoclusternode
set :autoclusternode.ranking_measure=Silhouette
set :autoclusternode.ranking_dataset=Training
set :autoclusternode.enable_silhouette_limit=true
set :autoclusternode.silhouette_limit=5

```

Propriétés autoclusternode	Valeurs	Description de la propriété
evaluation	<i>champ</i>	<i>Remarque</i> : Noeud de classification automatique uniquement : Identifie le champ pour lequel une valeur d'importance sera calculée. Il peut aussi être utilisé pour identifier la façon dont une classe différencie la valeur de ce champ et, par conséquent, la façon dont le modèle pourra prédire ce champ.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	<i>entier</i>	Nombre de modèles devant figurer dans le rapport. Indiquez un entier compris entre 1 et 100.
enable_silhouette_limit	<i>booléen</i>	
silhouette_limit	<i>entier</i>	Entier compris entre 0 et 100.
enable_number_less_limit	<i>booléen</i>	
number_less_limit	<i>nombre</i>	Nombre réel compris entre 0,0 et 1,0.
enable_number_greater_limit	<i>booléen</i>	
number_greater_limit	<i>nombre</i>	Entier supérieur à 0.
enable_smallest_cluster_limit	<i>booléen</i>	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percent- age	<i>nombre</i>	
smallest_cluster_limit_count	<i>entier</i>	Entier supérieur à 0.
enable_largest_cluster_limit	<i>booléen</i>	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percent- age	<i>nombre</i>	
largest_cluster_limit_count	<i>entier</i>	
enable_smallest_largest_limit	<i>booléen</i>	
smallest_largest_limit	<i>nombre</i>	
enable_importance_limit	<i>booléen</i>	
importance_limit_condition	Greater_than Less_than	

Propriétés autoclusternode	Valeurs	Description de la propriété
importance_limit_greater_than	<i>nombre</i>	Entier compris entre 0 et 100.
importance_limit_less_than	<i>nombre</i>	Entier compris entre 0 et 100.
<algorithm>	<i>booléen</i>	Active ou désactive l'utilisation d'un algorithme particulier, par exemple : set :autoclusternode.kohonen=true
<algorithm>.<property>	<i>chaîne</i>	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, reportez-vous à la section Définition des propriétés de l'algorithme sur p. 202.

Propriétés de autonumericnode



Le noeud Numérisation automatique évalue et compare des modèles pour des résultats d'intervalle numérique continus par le biais de différentes méthodes. Le noeud fonctionne de la même manière que le noeud Classificateur automatique, vous permettant ainsi de choisir les algorithmes à utiliser et à tester avec différentes combinaisons d'options en un seul passage de modélisation. Les algorithmes pris en charge comprennent les réseaux neuraux, l'arborescence C&R Tree, CHAID, la régression linéaire, la régression linéaire généralisée et support vector machines (SVM). Les modèles peuvent être comparés selon la corrélation, l'erreur relative ou le nombre de variables utilisées. [Pour plus d'informations, reportez-vous à la section Noeud Numérisation automatique dans le chapitre 5 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create autonumericnode
set :autonumericnode.ranking_measure=Correlation
set :autonumericnode.ranking_dataset=Training
set :autonumericnode.enable_correlation_limit=true
set :autonumericnode.correlation_limit=0.8
set :autonumericnode.calculate_variable_importance=true
set :autonumericnode.neuralnetwork=true
set :autonumericnode.chaid=false
```

Propriétés autonumericnode	Valeurs	Description de la propriété
custom_fields	<i>booléen</i>	Si cette valeur est définie sur True (vrai), les paramètres de champ personnalisés seront utilisés à la place des paramètres du noeud Typer.
target	<i>champ</i>	Le noeud Numérisation automatique requiert un seul champ cible et un ou plusieurs champs d'entrée. Il est également possible de spécifier des champs de pondération et d'effectifs. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
inputs	<i>[champ1 ... champ2]</i>	
partition	<i>champ</i>	
use_frequency	<i>booléen</i>	

Propriétés autonumericnode	Valeurs	Description de la propriété
frequency_field	champ	
use_weight	booléen	
weight_field	champ	
use_partitioned_data	booléen	Si un champ de partition est défini, seules les données d'apprentissage sont utilisées pour la création du modèle.
ranking_measure	Correlation NumberOfFields	
ranking_dataset	Test Training	
number_of_models	entier	Nombre de modèles à inclure dans le nugget de modèle. Indiquez un entier compris entre 1 et 100.
calculate_variable_importance	booléen	
enable_correlation_limit	booléen	
correlation_limit	entier	
enable_number_of_fields_limit	booléen	
number_of_fields_limit	entier	
enable_relative_error_limit	booléen	
relative_error_limit	entier	
enable_model_build_time_limit	booléen	
model_build_time_limit	entier	
enable_stop_after_time_limit	booléen	
stop_after_time_limit	entier	
stop_if_valid_model	booléen	
<algorithm>	booléen	Active ou désactive l'utilisation d'un algorithme particulier, par exemple : set :autonumericnode.chaid=true
<algorithm>.<property>	chaîne	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, reportez-vous à la section Définition des propriétés de l'algorithme sur p. 202.

Propriétés de bayesnetnode



Le noeud Réseau Bayésien permet de créer un modèle de probabilité en combinant les preuves observées et enregistrées avec les connaissances réelles pour établir la probabilité des occurrences. Le noeud est axé sur le Tree Augmented Naïve Bayes (TAN) et sur les réseaux Couverture de Markov qui servent principalement à la classification. [Pour plus d'informations, reportez-vous à la section Noeud Réseau Bayésien dans le chapitre 7 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create bayesnetnode
set :bayesnetnode.continue_training_existing_model = True
set :bayesnetnode.structure_type = MarkovBlanket
```

```

set :bayesnetnode.use_feature_selection = True
# Expert tab
set :bayesnetnode.mode = Expert
set :bayesnetnode.all_probabilities = True
set :bayesnetnode.independence = Pearson

```

Propriétés bayesnetnode	Valeurs	Description de la propriété
inputs	<i>[champ1 ... champN]</i>	Les modèles de réseau Bayésien utilisent un seul champ cible et un ou plusieurs champs d'entrée. Les champs continus sont automatiquement mis en intervalles. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue_training_existing_model	<i>booléen</i>	
structure_type	TAN MarkovBlanket	Sélectionne la structure à utiliser lors de la création du réseau Bayésien.
use_feature_selection	<i>booléen</i>	
parameter_learning_method	Likelihood Bayes	Spécifie la méthode utilisée pour estimer les tableaux de probabilités conditionnelles entre les noeuds où les valeurs des parent sont connues.
mode	Expert Simple	
missing_values	<i>booléen</i>	
all_probabilities	<i>booléen</i>	
independence	Likelihood Pearson	Spécifie la méthode utilisée pour déterminer si des observations par paire sur deux variables sont indépendantes l'une de l'autre.
significance_level	<i>nombre</i>	Spécifie la valeur de césure permettant de déterminer l'indépendance.
maximal_conditioning_set	<i>nombre</i>	Définit le nombre maximum de variables de conditionnement à utiliser pour les essais d'indépendance.
inputs_always_selected	<i>[champ1 ... champN]</i>	Spécifie les champs de l'ensemble de données qui doivent toujours être utilisés lors de la création du réseau Bayésien. <i>Remarque</i> : Le champ cible est toujours sélectionné.
maximum_number_inputs	<i>nombre</i>	Spécifie le nombre maximum de champs d'entrée à utiliser pour créer le réseau Bayésien.
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés de c50node



Le noeud C5.0 crée un arbre décision ou un ensemble de règles. Le fonctionnement de ce modèle repose sur un découpage de l'échantillon basé sur le champ qui fournit le gain d'informations le plus important à chaque niveau. Le champ cible doit être catégoriel. Les divisions multiples en plus de deux sous-groupes sont autorisées. [Pour plus d'informations, reportez-vous à la section Noeud C5.0 dans le chapitre 6 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create c50node
# "Model" tab
set :c50node.use_model_name = False
set :c50node.model_name = "C5_Drug"
set :c50node.use_partitioned_data = True
set :c50node.output_type = DecisionTree
set :c50node.use_xval = True
set :c50node.xval_num_folds = 3
set :c50node.mode = Expert
set :c50node.favor = Generality
set :c50node.min_child_records = 3
# "Costs" tab
set :c50node.use_costs = True
set :c50node.costs = [{"drugA" "drugX" 2}]
```

Propriétés c50node	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles C50 utilisent un seul champ cible et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
output_type	DecisionTree RuleSet	
group_symbolics	<i>booléen</i>	
use_boost	<i>booléen</i>	
boost_num_trials	<i>nombre</i>	
use_xval	<i>booléen</i>	
xval_num_folds	<i>nombre</i>	
mode	Simple Expert	
favor	Accuracy Generality	Privilégiez la précision ou la généralité.
expected_noise	<i>nombre</i>	
min_child_records	<i>nombre</i>	
pruning_severity	<i>nombre</i>	
use_costs	<i>booléen</i>	
costs	<i>structurées</i>	Il s'agit d'une propriété structurée.
use_winning	<i>booléen</i>	

Propriétés c50node	Valeurs	Description de la propriété
use_global_pruning	booléen	Activé (True) par défaut.
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés de carmanode



Le modèle CARMA extrait un ensemble de règles des données sans que vous ayez à définir les champs d'entrée ou les champs cible. Au contraire du noeud Apriori le noeud CARMA offre des paramètres de création pour la prise en charge de la règle (à la fois pour les antécédents et les conséquences), plutôt qu'une simple prise en charge d'antécédents. Cela signifie que les règles générées peuvent être utilisées dans un grand nombre d'applications, par exemple pour rechercher une liste des produits ou des services (antécédents) dont la conséquence correspond à l'élément que vous souhaitez promouvoir à l'occasion de cette période de congés. [Pour plus d'informations, reportez-vous à la section Noeud CARMA dans le chapitre 12 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create carmanode
# "Fields" tab
set :carmanode.custom_fields = True
set :carmanode.use_transactional_data = True
set :carmanode.inputs = ['BP' 'Cholesterol' 'Drug']
set :carmanode.partition = Test
# "Model" tab
set :carmanode.use_model_name = False
set :carmanode.model_name = "age_bp_drug"
set :carmanode.use_partitioned_data = False
set :carmanode.min_supp = 10.0
set :carmanode.min_conf = 30.0
set :carmanode.max_size = 5
# Expert Options
set :carmanode.mode = Expert
#help set :carmanode.exclude_simple = True
set :carmanode.use_pruning = True
set :carmanode.pruning_value = 300
set :carmanode.vary_support = True
set :carmanode.estimated_transactions = 30
```

```
set :carmanode.rules_without_antecedents = True
```

Propriétés carmanode	Valeurs	Description de la propriété
inputs	<i>[champ1 ... champn]</i>	Les modèles CARMA utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
id_field	<i>champ</i>	Champ utilisé en tant que champ d'ID pour la création du modèle.
contiguous	<i>booléen</i>	Permet d'indiquer si les ID du champ ID sont contigus.
use_transactional_data	<i>booléen</i>	
content_field	<i>champ</i>	
min_supp	<i>nombre (pourcentage)</i>	Est lié à la prise en charge de la règle plutôt qu'à celle des antécédents. La valeur par défaut est 20 %.
min_conf	<i>nombre (pourcentage)</i>	La valeur par défaut est 20 %.
max_size	<i>nombre</i>	La valeur par défaut est 10.
mode	Simple Expert	La valeur par défaut est Simple.
exclude_multiple	<i>booléen</i>	Exclut les règles comportant plusieurs conséquences. La valeur par défaut est False.
use_pruning	<i>booléen</i>	La valeur par défaut est False.
pruning_value	<i>nombre</i>	La valeur par défaut est 500.
vary_support	<i>booléen</i>	
estimated_transactions	<i>entier</i>	
rules_without_antecedents	<i>booléen</i>	

Propriétés de cartnode



Le noeud Arbre Classification et Regression, (C&R) Tree, génère un arbre décision qui vous permet de prévoir ou de classifier les observations futures. La méthode utilise la technique de partition récursive afin de diviser les données d'apprentissage en segments en réduisant l'index d'impureté à chaque étape, un noeud de l'arbre étant considéré comme "pur" si 100 % de ses observations appartiennent à une catégorie spécifique du champ cible. Les champs cible et les champs d'entrée peuvent être des champs d'intervalle numériques ou des champs catégoriels numériques (nominal, ordinal ou booléen). Toutes les divisions sont binaires (deux sous-groupes uniquement). [Pour plus d'informations, reportez-vous à la section Noeud Arbre C&RT dans le chapitre 6 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create cartnode
# "Fields" tab
set :cartnode.custom_fields = True
set :cartnode.target = 'Drug'
```



```

set :cartnode.inputs = ['Age' 'BP' 'Cholesterol']
# "Build Options" tab, 'Objective' panel
set :cartnode.model_output_type = InteractiveBuilder
set :cartnode.use_tree_directives = True
set :cartnode.tree_directives = ""Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""
# "Build Options" tab, 'Basics' panel
set :cartnode.prune_tree = False
set :cartnode.use_std_err_rule = True
set :cartnode.std_err_multiplier = 3.0
set :cartnode.max_surrogates = 7
# "Build Options" tab, 'Stopping Rules' panel
set :cartnode.use_percentage = True
set :cartnode.min_parent_records_pc = 5
set :cartnode.min_child_records_pc = 3
# "Build Options" tab, 'Costs & Priors' panel
set :cartnode.use_costs = True
set :cartnode.costs = [{"drugA" "drugX" 2}]
set :cartnode.priors = Custom
# les probabilités personnalisées doivent ajouter 1
set :cartnode.custom_priors = [{"drugA" 0.3}{ "drugX" 0.7}]
set :cartnode.adjust_priors = True
# "Build Options" tab, 'Advanced' panel
set :cartnode.min_impurity = 0.0003
set :cartnode.impurity_measure = Twoing
# "Model Options" tab
set :cartnode.use_model_name = False
set :cartnode.model_name = "Cart_Drug"

```

Propriétés cartnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles Arbre C&R requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ d'effectifs peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue_training_existing_model	<i>booléen</i>	
objective	Standard Boosting Bagging psm	psm est destinée aux ensembles de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>booléen</i>	

Propriétés cartnode	Valeurs	Description de la propriété
tree_directives	chaîne	Définissez les directives de développement de l'arbre. Vous pouvez placer les directives entre guillemets triples pour ne pas tenir compte des caractères d'insertion de ligne ou des guillemets doubles. Notez que les directives sont parfois très sensibles aux modifications, même mineures, apportées aux données ou aux options de modélisation. Elles peuvent ne pas s'étendre aux autres ensembles de données. Pour plus d'informations, reportez-vous à la section Directives de développement d'arbre dans le chapitre 6 dans <i>Noeuds de modélisation de IBM SPSS Modeller 15</i>.
use_max_depth	Default Custom	
max_depth	entier	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom
prune_tree	booléen	Elagage de l'arbre pour éviter le surajustement.
use_std_err	booléen	Utiliser la différence maximale pour le risque (dans Erreurs standard).
std_err_multiplier	nombre	Différence maximale.
max_surrogates	nombre	Nombre maximal de substitutions.
use_percentage	booléen	
min_parent_records_pc	nombre	
min_child_records_pc	nombre	
min_parent_records_abs	nombre	
min_child_records_abs	nombre	
use_costs	booléen	
costs	structurées	Propriété structurée utilisant la syntaxe : <code>{{drugA drugB 1.5}{drugA drugC 2.1}}</code> , où les arguments entre <code>{ }</code> sont les coûts prévus réels.
priors	Data Equal Custom	
custom_priors	structurées	Propriété structurée utilisant la syntaxe : <code>set :cartnode. custom_priors = [{ drugA 0.3 } { drugB 0.6 }]</code>
adjust_priors	booléen	
trails	nombre	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.

Propriétés cartnode	Valeurs	Description de la propriété
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	booléen	Appliquer le boosting aux ensembles de données très volumineux.
min_impurity	nombre	
impurity_measure	Gini Twoing Ordered	
train_pct	nombre	Ensemble de prévention de surajustement.
set_random_seed	booléen	Dupliquer l'option des résultats.
seed	nombre	
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés de chaidnode



Le noeud CHAID génère des arbres décision à l'aide des statistiques du Chi-deux pour identifier les séparations optimales. Contrairement aux noeuds C&R Tree et QUEST, CHAID peut générer des arbres non binaires, ce qui implique que certaines divisions possèdent plusieurs branches. Les champs cibles et les champs d'entrée peuvent être d'intervalle numérique (continu) ou catégoriques. La méthode Exhaustive CHAID correspond à une modification du CHAID qui examine plus en détail toutes les divisions possibles, mais dont les calculs sont plus longs. [Pour plus d'informations, reportez-vous à la section Noeud CHAID dans le chapitre 6 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create chaidnode
set :chaidnode.custom_fields = True
set :chaidnode.target = Drug
set :chaidnode.inputs = [Age Na K Cholesterol BP]
set :chaidnode.use_model_name = true
set :chaidnode.model_name = "CHAID"
set :chaidnode.method = Chaid
set :chaidnode.model_output_type = InteractiveBuilder
set :chaidnode.use_tree_directives = True
set :chaidnode.tree_directives = "Test"
set :chaidnode.mode = Expert
set :chaidnode.split_alpha = 0.03
set :chaidnode.merge_alpha = 0.04
set :chaidnode.chi_square = Pearson
set :chaidnode.use_percentage = True
set :chaidnode.min_parent_records_abs = 40
set :chaidnode.min_child_records_abs = 30
set :chaidnode.epsilon = 0.003
```

```
set :chaidnode.max_iterations = 75
set :chaidnode.split_merged_categories = true
set :chaidnode.bonferroni_adjustment = true
```

Propriétés chaidnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles CHAID requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ d'effectifs peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue_training_existing_model	<i>booléen</i>	
objective	Standard Boosting Bagging psm	psm est destinée aux ensembles de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>booléen</i>	
tree_directives	<i>chaîne</i>	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	<i>entier</i>	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom
use_percentage	<i>booléen</i>	
min_parent_records_pc	<i>nombre</i>	
min_child_records_pc	<i>nombre</i>	
min_parent_records_abs	<i>nombre</i>	
min_child_records_abs	<i>nombre</i>	
use_costs	<i>booléen</i>	
costs	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , où les arguments entre <code>{}</code> sont les coûts prévus réels.
trails	<i>nombre</i>	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	<i>booléen</i>	Appliquer le boosting aux ensembles de données très volumineux.
split_alpha	<i>nombre</i>	Niveau de signification pour la division.

Propriétés chaidnode	Valeurs	Description de la propriété
merge_alpha	nombre	Niveau de signification pour la fusion.
bonferroni_adjustment	booléen	Ajuster les valeurs de signification à l'aide de la méthode Bonferroni.
split_merged_categories	booléen	Autoriser une nouvelle division des catégories fusionnées.
chi_square	Pearson LR	Méthode utilisée pour calculer les statistiques du Khi-deux : le Khi-deux de Pearson ou le Khi-deux du rapport de vraisemblance
epsilon	nombre	Modification minimale dans les prévisions de fréquence de cellule.
max_iterations	nombre	Itérations maximales pour convergence.
set_random_seed	entier	
seed	nombre	
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	entier	

Propriétés de coxregnode



Le noeud de régression de Cox vous permet de créer un modèle de survie pour les données de durée jusqu'à l'événement en présence d'enregistrements censurés. Ce modèle produit une fonction de survie qui prédit la probabilité que l'événement en question se soit produit à un moment (t) pour des valeurs données des variables d'entrée. [Pour plus d'informations, reportez-vous à la section Noeud de Cox dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create coxregnode
set :coxregnode.survival_time = tenure
set :coxregnode.method = BackwardsStepwise
# Expert tab
set :coxregnode.mode = Expert
set :coxregnode.removal_criterion = Conditional
```

```
set :coxregnode.survival = True
```

Propriétés coxregnode	Valeurs	Description de la propriété
survival_time	<i>champ</i>	Les modèles de régression de Cox requièrent un champ unique contenant les durées de survie.
target	<i>champ</i>	Les modèles de régression de Cox requièrent un seul champ cible et un ou plusieurs champs d'entrée. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
method	Enter Stepwise BackwardsStepwise	
groups	<i>champ</i>	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	Exemple : set :coxregnode. custom_terms = ["BP*Sex" "BP" "Age"]
mode	Expert Simple	
max_iterations	<i>nombre</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	<i>nombre</i>	
probability_removal	<i>nombre</i>	
output_display	EachStep LastStep	
ci_enable	<i>booléen</i>	
ci_value	90 95 99	

Propriétés coxregnode	Valeurs	Description de la propriété
correlation	booléen	
display_baseline	booléen	
survival	booléen	
hazard	booléen	
log_minus_log	booléen	
one_minus_survival	booléen	
separate_line	champ	
value	nombre ou chaîne	Si aucune valeur n'est spécifiée pour un champ, l'option par défaut, « Moyenne » sera utilisée pour ce champ. Utilisation pour un champ numérique : coxnode.value = [{"age" "35.8"}] Utilisation pour un champ catégoriel : coxnode.value = [{"color" "pink"}]

Propriétés de decisionlistnode



Le noeud Liste de décision identifie les sous-groupes, ou les segments, qui présentent une probabilité plus élevée ou plus faible d'un résultat binaire donné par rapport à la population globale. Vous pouvez, par exemple, rechercher les clients qui ont une faible probabilité d'attrition ou ceux qui ont une plus forte probabilité de répondre favorablement à une campagne. Vous pouvez incorporer votre connaissance du marché dans le modèle en ajoutant vos propres segments personnalisés et en prévisualisant des modèles alternatifs côte à côte de façon à comparer les résultats. Les modèles Liste de décision se composent d'une liste de règles dans laquelle chaque règle présente une condition et un résultat. Les règles sont appliquées dans l'ordre et la première règle correspondante détermine le résultat. [Pour plus d'informations, reportez-vous à la section Liste de décision dans le chapitre 9 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create decisionlistnode
set :decisionlistnode.search_direction=Down
set :decisionlistnode.target_value=1
set :decisionlistnode.max_rules=4
set :decisionlistnode.min_group_size_pct = 15
```

Propriétés decisionlistnode	Valeurs	Description de la propriété
target	champ	Les modèles Liste de décision utilisent un seul champ cible et un ou plusieurs champs d'entrée. Un champ d'effectifs peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
model_output_type	Model InteractiveBuilder	

Propriétés decisionlistnode	Valeurs	Description de la propriété
search_direction	Up Down	Est lié à la recherche de segments ; où Up est l'équivalent de Forte probabilité et Down est l'équivalent de Faible probabilité.
target_value	chaîne	Si cette propriété n'est pas définie, la valeur True (vrai) est sélectionnée pour les champs booléens.
max_rules	entier	Nombre maximal de segments, sauf le reste.
min_group_size	entier	Taille minimale de segment.
min_group_size_pct	nombre	Taille minimale de segment, en pourcentage.
confidence_level	nombre	Seuil minimal dont dispose un champ d'entrée pour améliorer (augmenter) la probabilité d'une réponse, pour que son ajout à une définition de segment soit utile.
max_segments_per_rule	entier	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	nombre	
max_models_per_cycle	entier	Largeur de recherche des listes.
max_rules_per_cycle	entier	Largeur de recherche des règles de segment.
segment_growth	nombre	
include_missing	booléen	
final_results_only	booléen	
reuse_fields	booléen	Permet aux attributs (champs d'entrée qui apparaissent dans les règles) d'être réutilisés.
max_alternatives	entier	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés de discriminantnode



L'analyse discriminante crée des hypothèses plus strictes que la régression logistique mais peut constituer une alternative ou un complément précieux à une analyse de régression logistique lorsque ces hypothèses sont réunies. [Pour plus d'informations, reportez-vous à la section Noeud discriminant dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```

create discriminantnode
set :discriminantnode.target = custcat
set :discriminantnode.use_partitioned_data = False
set :discriminantnode.method = Stepwise

```

Propriétés discriminantnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles Discriminant requièrent un seul champ cible et un ou plusieurs champs d'entrée. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	<i>booléen</i>	Options Statistiques dans la boîte de dialogue Sortie avancée.
univariate_anovas	<i>booléen</i>	
box_m	<i>booléen</i>	
within_group_covariance	<i>booléen</i>	
within_groups_correlation	<i>booléen</i>	
separate_groups_covariance	<i>booléen</i>	
total_covariance	<i>booléen</i>	
fishers	<i>booléen</i>	
unstandardized	<i>booléen</i>	
casewise_results	<i>booléen</i>	Options Classification dans la boîte de dialogue Sortie avancée.
limit_to_first	<i>nombre</i>	La valeur par défaut est 10.
summary_table	<i>booléen</i>	
leave_one_classification	<i>booléen</i>	
combined_groups	<i>booléen</i>	
separate_groups_covariance	<i>booléen</i>	Option Matrices, Covariance par groupes distincts
territorial_map	<i>booléen</i>	
combined_groups	<i>booléen</i>	Option Nuage Groupes regroupés.
separate_groups	<i>booléen</i>	Option Nuage Groupes distincts
summary_of_steps	<i>booléen</i>	
F_pairwise	<i>booléen</i>	

Propriétés discriminantnode	Valeurs	Description de la propriété
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	<i>nombre</i>	
criteria	UseValue UseProbability	
F_value_entry	<i>nombre</i>	La valeur par défaut est 3,84.
F_value_removal	<i>nombre</i>	La valeur par défaut est 2,71.
probability_entry	<i>nombre</i>	La valeur par défaut est 0.05.
probability_removal	<i>nombre</i>	La valeur par défaut est 0.10.
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés de factornode



Le noeud ACP/Analyse factorielle propose des techniques de factorisation puissantes qui vous permettent de réduire la complexité de vos données. L'analyse en composantes principales (ACP) recherche les combinaisons linéaires des champs d'entrée qui permettent de capturer au mieux la variance dans l'ensemble de champs, où les composantes sont orthogonales (perpendiculaires) les unes par rapport aux autres. L'analyse factorielle a pour but d'identifier les facteurs sous-jacents qui expliquent la tendance des corrélations dans un ensemble de champs observés. Quelle que soit l'approche choisie, le but consiste à trouver un nombre limité de champs dérivés récapitulant les informations contenues dans l'ensemble de champs d'origine. [Pour plus d'informations, reportez-vous à la section Noeud ACP/Analyse factorielle dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create factornode
# "Fields" tab
set :factornode.custom_fields = True
set :factornode.inputs = ['BP' 'Na' 'K']
set :factornode.partition = Test
# "Model" tab
set :factornode.use_model_name = True
set :factornode.model_name = "Factor_Age"
set :factornode.use_partitioned_data = False
set :factornode.method = GLS
# Expert options
set :factornode.mode = Expert
set :factornode.complete_records = true
set :factornode.matrix = Covariance
set :factornode.max_iterations = 30
set :factornode.extract_factors = ByFactors
```

```

set :factornode.min_eigenvalue = 3.0
set :factornode.max_factor = 7
set :factornode.sort_values = True
set :factornode.hide_values = True
set :factornode.hide_below = 0.7
# "Rotation" section
set :factornode.rotation = DirectOblimin
set :factornode.delta = 0.3
set :factornode.kappa = 7.0

```

Propriétés factornode	Valeurs	Description de la propriété
inputs	[<i>champ1 ... champN</i>]	Les modèles APC/Facteur utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	<i>nombre</i>	
complete_records	<i>booléen</i>	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	<i>nombre</i>	
max_factor	<i>nombre</i>	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	<i>nombre</i>	Si vous sélectionnez DirectOblimin comme type de données de rotation, vous pouvez indiquer une valeur de delta. Si vous ne le faites pas, la valeur par défaut de delta est utilisée.
kappa	<i>nombre</i>	Si vous sélectionnez Promax comme type de données de rotation, vous pouvez indiquer une valeur de kappa. Si vous ne le faites pas, la valeur par défaut de kappa est utilisée.
sort_values	<i>booléen</i>	

Propriétés factornode	Valeurs	Description de la propriété
hide_values	booléen	
hide_below	nombre	

Propriétés de featureselectionnode



Le noeud Sélection de fonction filtre les champs d'entrée en vue de leur suppression, en fonction d'un ensemble de critères donné (tel que le pourcentage de valeurs manquantes) ; il classe ensuite les entrées restantes selon leur importance par rapport à la cible indiquée. Si l'on prend, par exemple, un ensemble de données comportant des centaines d'entrées potentielles, quelles sont celles susceptibles d'être les plus utiles dans la modélisation des résultats de patients ? [Pour plus d'informations, reportez-vous à la section Noeud Sélection de fonction dans le chapitre 4 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create featureselectionnode
set :featureselectionnode.screen_single_category=true
set :featureselectionnode.max_single_category=95
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.unimportant_below = 0.8
set :featureselectionnode.important_above = 0.9
set :featureselectionnode.important_label = "Check Me Out!"
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
```

Pour un exemple plus détaillé de création et d'application d'un modèle Sélection de fonction, reportez-vous à [Exemple de script autonome : Génération d'un modèle Sélection de fonction dans le chapitre 2 sur p. 14](#).

Propriétés featureselectionnode	Valeurs	Description de la propriété
target	champ	Les modèles Sélection de fonction classent les variables indépendantes par rapport à la cible spécifiée. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
screen_single_category	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs qui comportent, par rapport au nombre total d'enregistrements, trop d'enregistrements relatifs à une même catégorie.
max_single_category	nombre	Spécifie le seuil utilisé quand screen_single_category est True.

Propriétés featureselectionnode	Valeurs	Description de la propriété
screen_missing_values	<i>booléen</i>	Si cette propriété est définie sur True (vrai), elle filtre les champs qui comportent un nombre trop important de valeurs manquantes, exprimé en pourcentage du nombre total d'enregistrements.
max_missing_values	<i>nombre</i>	
screen_num_categories	<i>booléen</i>	Si cette propriété est définie sur True (vrai), elle filtre les champs comportant trop de catégories par rapport au nombre total d'enregistrements.
max_num_categories	<i>nombre</i>	
screen_std_dev	<i>booléen</i>	Si cette propriété est définie sur True, elle filtre les champs dont l'écart-type est inférieur ou égal à la valeur minimale indiquée.
min_std_dev	<i>nombre</i>	
screen_coeff_of_var	<i>booléen</i>	Si cette propriété est définie sur True (vrai), elle filtre les champs dont le coefficient de la variance est inférieur ou égal à la valeur minimale spécifiée.
min_coeff_of_var	<i>nombre</i>	
criteria	Pearson Likelihood CramersV Lambda	Lors du classement des variables indépendantes catégorielles par rapport à une cible catégorielle, indique la mesure sur laquelle est basée la valeur d'importance.
unimportant_below	<i>nombre</i>	Indique les valeurs p du seuil, utilisées pour classer les variables comme étant importantes, marginales ou non significatives. Accepte des valeurs de 0,0 à 1,0.
important_above	<i>nombre</i>	Accepte des valeurs de 0,0 à 1,0.
unimportant_label	<i>chaîne</i>	Indique l'étiquette du classement non significatif.
marginal_label	<i>chaîne</i>	
important_label	<i>chaîne</i>	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	<i>booléen</i>	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs importants doivent être sélectionnés.
select_marginal	<i>booléen</i>	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs marginaux doivent être sélectionnés.

Propriétés featureselectionnode	Valeurs	Description de la propriété
select_unimportant	<i>booléen</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceLevel</code> , indique si les champs non importants doivent être sélectionnés.
importance_value	<i>nombre</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceValue</code> , indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 100.
top_n	<i>entier</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>TopN</code> , indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 1000.

Propriétés de **genlinnode**



La procédure Modèles linéaires généralisés développe le modèle linéaire général de sorte que la variable dépendante soit linéairement reliée aux facteurs et covariables via une fonction de lien précise. En outre, le modèle permet à la variable dépendante de suivre une distribution non normale. Il couvre les fonctionnalités d'un grand nombre de modèles statistiques, notamment le modèle de régression linéaire, le modèle de régression logistique, le modèle log-linéaire pour les données d'effectif et le modèle de survie avec censure par intervalle. [Pour plus d'informations, reportez-vous à la section Noeud Modèles linéaires généralisés dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create genlinnode
set :genlinnode.model_type = MainAndAllTwoWayEffects
set :genlinnode.offset_type = Variable
set :genlinnode.offset_field = Claimant
```

Propriétés genlinnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles linéaires généralisés requièrent un seul champ cible, qui doit être un champ nominal ou booléen, et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
use_weight	<i>booléen</i>	
weight_field	<i>champ</i>	Le champ peut uniquement être de type continu.
target_represents_trials	<i>booléen</i>	
trials_type	Variable FixedValue	
trials_field	<i>champ</i>	Le champ est de type Continu, Booléen ou Ordinal.
trials_number	<i>nombre</i>	La valeur par défaut est 10.

Propriétés genlinnode	Valeurs	Description de la propriété
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	<i>champ</i>	Le champ peut uniquement être de type continu.
offset_value	<i>nombre</i>	Doit être un nombre réel.
base_category	Last First	
include_intercept	<i>booléen</i>	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS : Gaussienne inverse. NEGBIN : Binomiale négative.
negbin_para_type	Specify Estimate	
negbin_parameter	<i>nombre</i>	La valeur par défaut est 1. Doit contenir un nombre réel non négatif.
tweedie_parameter	<i>nombre</i>	
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG : Log-log complémentaire. LOGC : complément log. NEGBIN : Binomiale négative. NLOGLOG : Log-log négatif. CUMCAUCHIT : Cauchit cumulé. CUMCLOGLOG : Log-log complémentaire cumulé. CUMLOGIT : Logit cumulé. CUMNLOGLOG : Log-log négatif cumulé. CUMPROBIT : Probit cumulé.
power	<i>nombre</i>	La valeur doit être un nombre réel autre que zéro.
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	<i>nombre</i>	La valeur par défaut est 1 ; seuls les entiers positifs sont autorisés.
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	<i>nombre</i>	La valeur par défaut est 1 ; elle doit être supérieure à 0.

Propriétés genlinnode	Valeurs	Description de la propriété
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	<i>nombre</i>	La valeur par défaut est 100 ; entiers non négatifs uniquement.
max_step_halving	<i>nombre</i>	La valeur par défaut est 5 ; entiers positifs uniquement.
check_separation	<i>booléen</i>	
start_iteration	<i>nombre</i>	La valeur par défaut est 20 ; seuls les entiers positifs sont autorisés.
estimates_change	<i>booléen</i>	
estimates_change_min	<i>nombre</i>	La valeur par défaut est 1E-006 ; seuls les nombres positifs sont autorisés.
estimates_change_type	Absolute Relative	
loglikelihood_change	<i>booléen</i>	
loglikelihood_change_min	<i>nombre</i>	Seuls les nombres positifs sont autorisés.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	<i>booléen</i>	
hessian_convergence_min	<i>nombre</i>	Seuls les nombres positifs sont autorisés.
hessian_convergence_type	Absolute Relative	
case_summary	<i>booléen</i>	
contrast_matrices	<i>booléen</i>	
descriptive_statistics	<i>booléen</i>	
estimable_functions	<i>booléen</i>	
model_info	<i>booléen</i>	
iteration_history	<i>booléen</i>	
goodness_of_fit	<i>booléen</i>	
print_interval	<i>nombre</i>	La valeur par défaut est 1 ; il doit s'agir d'un entier positif.
model_summary	<i>booléen</i>	
lagrange_multiplieur	<i>booléen</i>	
parameter_estimates	<i>booléen</i>	
include_exponential	<i>booléen</i>	
covariance_estimates	<i>booléen</i>	
correlation_estimates	<i>booléen</i>	
analysis_type	TypeI TypeIII TypeIAndTypeIII	
statistics	Wald LR	
citype	Wald Profile	
tolerancelevel	<i>nombre</i>	La valeur par défaut est 0,0001.

Propriétés genlnode	Valeurs	Description de la propriété
confidence_interval	<i>nombre</i>	La valeur par défaut est 95.
loglikelihood_function	Full Kernel	
singularity_tolerance	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
value_order	Ascending Descending DataOrder	
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés glmmnode



Un modèle mixte linéaire généralisé (MMLG) élargit le modèle linéaire de sorte que la cible puisse avoir une distribution non normale, qu'elle soit liée linéairement aux facteurs et covariables via une fonction de lien spécifiée, et que les observations puissent être corrélées. Les modèles mixtes linéaires généralisés couvrent une large variété de modèles, depuis les modèles de régression linéaire simple aux modèles multi-niveaux complexes destinés aux données longitudinales non normales. [Pour plus d'informations, reportez-vous à la section Noeud MMLG dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Propriétés glmmnode	Valeurs	Description de la propriété
residual_subject_spec	<i>structurées</i>	La combinaison des valeurs des champs catégoriels spécifiés qui définissent de manière unique les sujets dans l'ensemble de données.
repeated_measures	<i>structurées</i>	Champs à utiliser pour identifier les observations répétées.
residual_group_spec	[<i>champ1 ... champN</i>]	Champs qui définissent les ensembles indépendants des paramètres de covariance d'effets répétés.
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Spécifie la structure de la covariance des résidus.
custom_target	<i>booléen</i>	Indique s'il faut utiliser la cible définie dans le noeud en amont (<i>false</i>) ou la cible personnalisée spécifiée par <i>target_field</i> (<i>true</i>).

Propriétés glmmnode	Valeurs	Description de la propriété
target_field	<i>champ</i>	Champ à utiliser comme cible si <code>custom_target</code> est true.
use_trials	<i>booléen</i>	Indique si un champ supplémentaire ou une valeur spécifiant le nombre d'essais doit être utilisé lorsque la réponse cible est un nombre d'événements se produisant dans un ensemble d'essais. La valeur par défaut est false.
use_field_or_value	Field Value	Indique si un champ (par défaut) ou une valeur est utilisé pour spécifier le nombre d'essais.
trials_field	<i>champ</i>	Champ à utiliser pour spécifier le nombre d'essais.
trials_value	<i>entier</i>	Valeur à utiliser pour spécifier le nombre d'essais. Si spécifiée, la valeur minimum est 1.
use_custom_target_reference	<i>booléen</i>	Indique si la catégorie de référence personnalisée doit être utilisée pour une cible catégorielle. La valeur par défaut est false.
target_reference_value	<i>chaîne</i>	Catégorie de référence à utiliser si <code>use_custom_target_reference</code> est true.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	Modèles standard pour la distribution des valeurs de la cible. Choisissez Custom pour spécifier une distribution à partir de la liste fournie par <code>target_distribution</code> .
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	Distribution des valeurs de la cible quand <code>dist_link_combination</code> est Custom.
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	Fonction de lien pour associer les valeurs cibles aux variables indépendantes.
link_function_param	<i>nombre</i>	Valeur de paramètre de fonction de lien à utiliser. S'applique uniquement si <code>link_function</code> est POWER.
use_predefined_inputs	<i>booléen</i>	Indique si les champs d'effets fixes doivent être ceux définis en amont comme champs d'entrée (true) ou ceux de <code>fixed_effects_list</code> (false). La valeur par défaut est false.

Propriétés glmmnode	Valeurs	Description de la propriété
fixed_effects_list	<i>structurées</i>	Si <i>use_predefined_inputs</i> est <i>false</i> , spécifie les champs d'entrée à utiliser comme champs d'effets fixes.
use_intercept	<i>booléen</i>	Si <i>true</i> (par défaut), inclut la constante dans le modèle.
random_effects_list	<i>structurées</i>	Liste des champs à spécifier comme effets aléatoires.
regression_weight_field	<i>champ</i>	Champ à utiliser comme champ de pondération d'analyse.
use_offset	None offset_value offset_field	Indique comment le décalage est spécifié. La valeur <i>None</i> signifie qu'aucun décalage n'est spécifié.
offset_value	<i>nombre</i>	Valeur à utiliser pour le décalage si <i>use_offset</i> est défini sur <i>offset_value</i> .
offset_field	<i>champ</i>	Champ à utiliser pour la valeur de décalage si <i>use_offset</i> est défini sur <i>offset_field</i> .
target_category_order	Ascending Descending Data	Ordre de tri des cibles catégorielles. La valeur <i>Data</i> spécifie qu'il faut utiliser l'ordre de tri trouvé dans les données. La valeur par défaut est <i>Ascending</i> .
inputs_category_order	Ascending Descending Data	Ordre de tri pour les variables indépendantes catégorielles. La valeur <i>Data</i> spécifie qu'il faut utiliser l'ordre de tri trouvé dans les données. La valeur par défaut est <i>Ascending</i> .
max_iterations	<i>entier</i>	Nombre maximum d'itérations que l'algorithme effectuera. Un entier positif ; la valeur par défaut est 100.
confidence_level	<i>entier</i>	Niveau de confiance utilisé pour calculer les estimations d'intervalle des coefficients de modèle. Un entier positif ; le nombre maximum est 100, la valeur par défaut est 95.
degrees_of_freedom_method	Fixed Varied	Spécifie comment les degrés de liberté sont calculés pour le test de signification.
test_fixed_effects_coefficients	Model Robust	Méthode de calcul de la matrice de covariance des estimations de paramètre.
use_model_name	<i>booléen</i>	Indique s'il faut spécifier un nom personnalisé pour le modèle (<i>true</i>) ou utiliser le nom généré par le système (<i>false</i>). La valeur par défaut est <i>false</i> .
model_name	<i>chaîne</i>	Si <i>use_model_name</i> est <i>true</i> , spécifie le nom du modèle à utiliser.
confidence	highest difference	Base de calcul de la valeur de confiance de scoring : probabilité prédite la plus élevée ou différence entre les probabilités prédites les plus élevées et les deuxièmes plus élevées.

Propriétés glmmnode	Valeurs	Description de la propriété
score_category_probabilities	<i>booléen</i>	Si true, produit des probabilités prédites pour les cibles catégorielles. La valeur par défaut est false.
max_categories	<i>entier</i>	Si <i>score_category_probabilities</i> est true, spécifie le nombre maximum de catégories à enregistrer.
score_propensity	<i>booléen</i>	Si true, produit des scores de propension pour les champs cibles booléens qui indiquent la probabilité du résultat « true » pour le champ.
emeans	<i>structure</i>	Pour chaque champ catégoriel de la liste des effets fixes, spécifie s'il faut produire des moyennes marginales estimées.
covariance_list	<i>structure</i>	Pour chaque champ continu de la liste des effets fixes, spécifie s'il faut utiliser la moyenne ou une valeur personnalisée lors du calcul des moyennes marginales estimées.
mean_scale	Original Transformed	Spécifie s'il faut calculer les moyennes marginales estimées en fonction de l'échelle originale de la cible (par défaut) ou de la transformation de la fonction de lien.
comparison_adjustment_method	LSD SEQBONFERRONI SEOSIDAK	Méthode d'ajustement à utiliser lors des tests d'hypothèse avec contrastes multiples.

Propriétés de kmeansnode



Le noeud K-Means classe l'ensemble de données dans différents groupes (ou classes). La méthode définit un nombre de classes fixe, affecte à plusieurs reprises des enregistrements à des classes et ajuste les centres de classe, jusqu'à ce que le modèle ne puisse plus être amélioré. Au lieu de tenter de prédire un résultat, le modèle *k*-means utilise un processus connu sous le nom d'apprentissage non supervisé pour découvrir des tendances dans l'ensemble de champs d'entrée. [Pour plus d'informations, reportez-vous à la section Noeud K-Means dans le chapitre 11 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create kmeansnode
# "Fields" tab
set :kmeansnode.custom_fields = True
set :kmeansnode.inputs = ['Cholesterol' 'BP' 'Drug' 'Na' 'K' 'Age']
# "Model" tab
set :kmeansnode.use_model_name = False
set :kmeansnode.model_name = "Kmeans_allinputs"
set :kmeansnode.num_clusters = 9
set :kmeansnode.gen_distance = True
set :kmeansnode.cluster_label = "Number"
set :kmeansnode.label_prefix = "Kmeans_"
```

```

set :kmeansnode.optimize = Speed
#"Expert" tab
set :kmeansnode.mode = Expert
set :kmeansnode.stop_on = Custom
set :kmeansnode.max_iterations = 10
set :kmeansnode.tolerance = 3.0
set :kmeansnode.encoding_value = 0.3

```

Propriétés kmeansnode	Valeurs	Description de la propriété
inputs	[<i>champ1</i> ... <i>champN</i>]	Les modèles K-means procèdent à une analyse des classes sur un ensemble de champs d'entrée, mais n'utilisent pas de champ cible. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
num_clusters	<i>nombre</i>	
gen_distance	<i>booléen</i>	
cluster_label	String Number	
label_prefix	<i>chaîne</i>	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	<i>nombre</i>	
tolerance	<i>nombre</i>	
encoding_value	<i>nombre</i>	
optimize	Speed Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.

Propriétés knnnode



Le noeud *k*-Voisin le plus proche (KNN) associe une nouvelle observation à la catégorie ou à la valeur des objets *k* les plus proches dans l'espace du prédicteur, où *k* est un entier. Les observations semblables sont proches l'une de l'autre et les observations dissemblables sont éloignées l'une de l'autre. [Pour plus d'informations, reportez-vous à la section Noeud KNN dans le chapitre 16 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```

create knnnode
# Objectives tab
set: knnnode.objective = Custom
# Settings tab - Neighbors panel
set: knnnode.automatic_k_selection = false
set: knnnode.fixed_k = 2

```

```

set: knnnode.weight_by_importance = True
# Settings tab - Analyze panel
set: knnnode.save_distances = True

```

Propriétés knnnode	Valeurs	Description de la propriété
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	<i>booléen</i>	
use_case_labels	<i>booléen</i>	Cochez la case pour activer l'option suivante.
case_labels_field	<i>champ</i>	
identify_focal_cases	<i>booléen</i>	Cochez la case pour activer l'option suivante.
focal_cases_field	<i>champ</i>	
automatic_k_selection	<i>booléen</i>	
fixed_k	<i>entier</i>	Activé uniquement si automatic_k_selection est False.
minimum_k	<i>entier</i>	Activé uniquement si automatic_k_selection est True.
maximum_k	<i>entier</i>	
distance_computation	Euclidean CityBlock	
weight_by_importance	<i>booléen</i>	
range_predictions	Mean Median	
perform_feature_selection	<i>booléen</i>	
forced_entry_inputs	[<i>champ1 ... champN</i>]	
stop_on_error_ratio	<i>booléen</i>	
number_to_select	<i>entier</i>	
minimum_change	<i>nombre</i>	
validation_fold_assign_by_field	<i>booléen</i>	
number_of_folds	<i>entier</i>	Activé uniquement si validation_fold_assign_by_field est False
set_random_seed	<i>booléen</i>	
random_seed	<i>nombre</i>	
folds_field	<i>champ</i>	Activé uniquement si validation_fold_assign_by_field est True
all_probabilities	<i>booléen</i>	
save_distances	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés de kohonennode



Le noeud Kohonen génère un type de réseau de neurones qui peut être utilisé pour classer les données en groupes distincts. Lorsque l'apprentissage du réseau est terminé, les enregistrements similaires doivent être regroupés dans la connexion de sortie, tandis que les enregistrements différents sont à l'opposé. Vous pouvez étudier le nombre d'observations capturées par chaque unité du nugget de modèle afin d'identifier les unités fortes. Vous pouvez ainsi vous faire une idée du nombre de classes approprié. [Pour plus d'informations, reportez-vous à la section Noeud Kohonen dans le chapitre 11 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create kohonennode
# "Model" tab
set :kohonennode.use_model_name = False
set :kohonennode.model_name = "Symbolic Cluster"
set :kohonennode.stop_on = Time
set :kohonennode.time = 1
set :kohonennode.set_random_seed = True
set :kohonennode.random_seed = 12345
set :kohonennode.optimize = Speed
# "Expert" tab
set :kohonennode.mode = Expert
set :kohonennode.width = 3
set :kohonennode.length = 3
set :kohonennode.decay_style = Exponential
set :kohonennode.phase1_neighborhood = 3
set :kohonennode.phase1_eta = 0.5
set :kohonennode.phase1_cycles = 10
set :kohonennode.phase2_neighborhood = 1
set :kohonennode.phase2_eta = 0.2
set :kohonennode.phase2_cycles = 75
```

Propriétés kohonennode	Valeurs	Description de la propriété
inputs	[<i>champ1</i> ... <i>champN</i>]	Les modèles Kohonen utilisent une liste de champs d'entrée, mais pas de cible. Les champs d'effectifs et de pondération ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue	<i>booléen</i>	
show_feedback	<i>booléen</i>	
stop_on	Default Time	
time	<i>nombre</i>	
optimize	Speed Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.
cluster_label	<i>booléen</i>	
mode	Simple Expert	

Propriétés kohonenode	Valeurs	Description de la propriété
width	<i>nombre</i>	
length	<i>nombre</i>	
decay_style	Linear Exponential	
phase1_neighborhood	<i>nombre</i>	
phase1_eta	<i>nombre</i>	
phase1_cycles	<i>nombre</i>	
phase2_neighborhood	<i>nombre</i>	
phase2_eta	<i>nombre</i>	
phase2_cycles	<i>nombre</i>	

Propriétés de linearnode



Les modèles de régression linéaire prédisent une cible continue en fonction de relations linéaires entre la cible et une ou plusieurs variables prédites. [Pour plus d'informations, reportez-vous à la section Modèles linéaires dans le chapitre 10 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create linearnode
# Onglet Options de création - volet Objectifs
set: linearnode.objective = Standard
# Onglet Options de création - volet Sélection de modèle
set: linearnode.model_selection = BestSubsets
set: linearnode.criteria_best_subsets = ASE
# Onglet Options de création - volet Ensembles
set: linearnode.combining_rule_categorical = HighestMeanProbability
```

Propriétés de linearnode	Valeurs	Description de la propriété
target	<i>champ</i>	Spécifie un champ cible unique.
inputs	[<i>champ1 ... champN</i>]	Champs prédicteurs utilisés par le modèle.
continue_training_existing_model	<i>Booléen</i>	
objective	Standard Bagging Boosting psm	psm est destinée aux ensembles de données très volumineux et nécessite une connexion à un serveur.
use_auto_data_preparation	<i>Booléen</i>	
confidence_level	<i>nombre</i>	
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquared ASE	

Propriétés de linearnode	Valeurs	Description de la propriété
probability_entry	<i>nombre</i>	
probability_removal	<i>nombre</i>	
use_max_effects	<i>Booléen</i>	
max_effects	<i>nombre</i>	
use_max_steps	<i>Booléen</i>	
max_steps	<i>nombre</i>	
criteria_best_subsets	AICC AdjustedRSquared ASE	
combining_rule_continuous	Mean Median	
component_models_n	<i>nombre</i>	
use_random_seed	<i>Booléen</i>	
random_seed	<i>nombre</i>	
use_custom_model_name	<i>Booléen</i>	
custom_model_name	<i>chaîne</i>	
use_custom_name	<i>Booléen</i>	
custom_name	<i>chaîne</i>	
tooltip	<i>chaîne</i>	
keywords	<i>chaîne</i>	
annotation	<i>chaîne</i>	

Propriétés de logregnode



La régression logistique est une technique statistique de classification des enregistrements sur la base des valeurs des champs d'entrée. Excepté le fait qu'elle utilise un champ cible catégoriel et non pas numérique, cette régression est similaire à la régression linéaire. [Pour plus d'informations, reportez-vous à la section Noeud Logistique dans le chapitre 10 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple multinomial

```
create logregnode
# "Fields" tab
set :logregnode.custom_fields = True
set :logregnode.target = 'Drug'
set :logregnode.inputs = ['BP' 'Cholesterol' 'Age']
set :logregnode.partition = Test
# "Model" tab
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Drug"
set :logregnode.use_partitioned_data = True
set :logregnode.method = Stepwise
set :logregnode.logistic_procedure = Multinomial
set :logregnode.multinomial_base_category = BP
set :logregnode.model_type = FullFactorial
set :logregnode.custom_terms = [{BP Sex}{Age}{Na K}]
set :logregnode.include_constant = False
```

```

# "Expert" tab
set :logregnode.mode = Expert
set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# "Convergence..." section
set :logregnode.max_iterations = 50
set :logregnode.max_steps = 3
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
set :logregnode.delta = 0.03
# "Output..." section
set :logregnode.summary = True
set :logregnode.likelihood_ratio = True
set :logregnode.asymptotic_correlation = True
set :logregnode.goodness_fit = True
set :logregnode.iteration_history = True
set :logregnode.history_steps = 3
set :logregnode.parameters = True
set :logregnode.confidence_interval = 90
set :logregnode.asymptotic_covariance = True
set :logregnode.classification_table = True
# "Stepping" options
set :logregnode.min_terms = 7
set :logregnode.use_max_terms = true
set :logregnode.max_terms = 10
set :logregnode.probability_entry = 3
set :logregnode.probability_removal = 5
set :logregnode.requirements = Containment

```

Exemple binomial

```

create logregnode
# "Fields" tab
set :logregnode.custom_fields = True
set :logregnode.target = 'Cholesterol'
set :logregnode.inputs = ['BP' 'Drug' 'Age']
set :logregnode.partition = Test
# "Model" tab
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Cholesterol"
set :logregnode.multinomial_base_category = BP
set :logregnode.use_partitioned_data = True
set :logregnode.binomial_method = Forwards
set :logregnode.logistic_procedure = Binomial
set :logregnode.binomial_categorical_input = Sex
set :logregnode.binomial_input_contrast.Sex = Simple
set :logregnode.binomial_input_category.Sex = Last
set :logregnode.include_constant = False
# "Expert" tab
set :logregnode.mode = Expert

```

```

set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# "Convergence..." section
set :logregnode.max_iterations = 50
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
# "Output..." section
set :logregnode.binomial_output_display = at_each_step
set :logregnode.binomial_goodness_fit = True
set :logregnode.binomial_iteration_history = True
set :logregnode.binomial_parameters = True
set :logregnode.binomial_ci_enable = True
set :logregnode.binomial_ci = 85
# "Stepping" options
set :logregnode.binomial_removal_criterion = LR
set :logregnode.binomial_probability_removal = 0.2

```

Propriétés logregnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles de régression logistique requièrent un seul champ cible et un ou plusieurs champs d'entrée. Les champs d'effectifs et de pondération ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
logistic_procedure	Binomial Multinomial	
include_constant	<i>booléen</i>	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	Lorsque le type de modèle indiqué est FullFactorial, les méthodes d'analyse pas à pas ne sont pas exécutées, même si elles sont spécifiées. La méthode utilisée sera la méthode Enter. Si le type de modèle est paramétré sur Custom, mais qu'aucun champ personnalisé n'est indiqué, un modèle Effets principaux est créé.
custom_terms	[<i>{BP Sex}{BP}{Age}</i>]	Exemple : set :logregnode. custom_terms = [<i>{Na} {K} {Na K}</i>]

Propriétés logregnode	Valeurs	Description de la propriété
multinomial_base_category	<i>chaîne</i>	Indique le mode de détermination de la catégorie de référence.
binomial_categorical_input	<i>chaîne</i>	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	Propriété saisie pour l'entrée catégorielle qui indique la façon dont le contraste est déterminé. Syntaxe : NODE.binomial_input_contrast.FIELD-NAME
binomial_input_category	First Last	Propriété saisie pour l'entrée catégorielle qui indique la façon dont la catégorie de référence est déterminée. Syntaxe : NODE.binomial_input_category.FIELD-NAME
scale	None UserDefined Pearson Deviance	
scale_value	<i>nombre</i>	
all_probabilities	<i>booléen</i>	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	<i>nombre</i>	
use_max_terms	<i>booléen</i>	
max_terms	<i>nombre</i>	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	<i>nombre</i>	
probability_removal	<i>nombre</i>	
binomial_probability_entry	<i>nombre</i>	
binomial_probability_removal	<i>nombre</i>	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	<i>nombre</i>	
max_steps	<i>nombre</i>	

Propriétés logregnode	Valeurs	Description de la propriété
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	<i>nombre</i>	
iteration_history	<i>booléen</i>	
history_steps	<i>nombre</i>	
summary	<i>booléen</i>	
likelihood_ratio	<i>booléen</i>	
asymptotic_correlation	<i>booléen</i>	
goodness_fit	<i>booléen</i>	
parameters	<i>booléen</i>	
confidence_interval	<i>nombre</i>	
asymptotic_covariance	<i>booléen</i>	
classification_table	<i>booléen</i>	
stepwise_summary	<i>booléen</i>	
info_criteria	<i>booléen</i>	
monotonicity_measures	<i>booléen</i>	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	<i>booléen</i>	
binomial_parameters	<i>booléen</i>	
binomial_iteration_history	<i>booléen</i>	
binomial_classification_plots	<i>booléen</i>	
binomial_ci_enable	<i>booléen</i>	
binomial_ci	<i>nombre</i>	
binomial_residual	outliers all	
binomial_residual_enable	<i>booléen</i>	
binomial_outlier_threshold	<i>nombre</i>	
binomial_classification_cutoff	<i>nombre</i>	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	

Propriétés de *neuralnetnode*

Avertissement : Une nouvelle version du noeud de modélisation Réseau de neurones, avec des caractéristiques améliorées, est disponible dans cette version et est décrite dans la section suivante (*neuralnetwork*). Bien que vous soyez encore en mesure de créer et d'évaluer un modèle avec la version précédente, nous vous conseillons de procéder à la mise à jour de vos scripts afin d'utiliser la nouvelle version. Les détails de la version précédentes sont conservées ici pour référence.

Exemple

```
create neuralnetnode
# "Fields" tab
set :neuralnetnode.custom_fields = True
set :neuralnetnode.targets = ['Drug']
set :neuralnetnode.inputs = ['Age' 'Na' 'K' 'Cholesterol' 'BP']
# "Model" tab
set :neuralnetnode.use_partitioned_data = True
set :neuralnetnode.method = Dynamic
set :neuralnetnode.train_pct = 30
set :neuralnetnode.set_random_seed = True
set :neuralnetnode.random_seed = 12345
set :neuralnetnode.stop_on = Time
set :neuralnetnode.accuracy = 95
set :neuralnetnode.cycles = 200
set :neuralnetnode.time = 3
set :neuralnetnode.optimize = Speed
# "Multiple Method Expert Options" section
set :neuralnetnode.m_topologies = "5 30 5; 2 20 3, 1 10 1"
set :neuralnetnode.m_non_pyramids = False
set :neuralnetnode.m_persistence = 100
```

Propriétés <i>neuralnetnode</i>	Valeurs	Description de la propriété
targets	[<i>champ1</i> ... <i>champN</i>]	Le noeud Réseau de neurones attend un ou plusieurs champs cible et un ou plusieurs champs d'entrée. Les champs d'effectifs et de pondération sont ignorés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	<i>booléen</i>	
train_pct	<i>nombre</i>	
set_random_seed	<i>booléen</i>	
random_seed	<i>nombre</i>	

Propriétés neuralnetnode	Valeurs	Description de la propriété
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	Mode d'arrêt.
accuracy	<i>nombre</i>	Précision d'arrêt.
cycles	<i>nombre</i>	Cycles d'apprentissage.
time	<i>nombre</i>	Durée d'apprentissage (minutes).
continue	<i>booléen</i>	
show_feedback	<i>booléen</i>	
binary_encode	<i>booléen</i>	
use_last_model	<i>booléen</i>	
gen_logfile	<i>booléen</i>	
logfile_name	<i>chaîne</i>	
alpha	<i>nombre</i>	
initial_eta	<i>nombre</i>	
high_eta	<i>nombre</i>	
low_eta	<i>nombre</i>	
eta_decay_cycles	<i>nombre</i>	
hid_layers	One Two Three	
hl_units_one	<i>nombre</i>	
hl_units_two	<i>nombre</i>	
hl_units_three	<i>nombre</i>	
persistence	<i>nombre</i>	
m_topologies	<i>chaîne</i>	
m_non_pyramids	<i>booléen</i>	
m_persistence	<i>nombre</i>	
p_hid_layers	One Two Three	
p_hl_units_one	<i>nombre</i>	
p_hl_units_two	<i>nombre</i>	
p_hl_units_three	<i>nombre</i>	
p_persistence	<i>nombre</i>	
p_hid_rate	<i>nombre</i>	
p_hid_pers	<i>nombre</i>	
p_inp_rate	<i>nombre</i>	
p_inp_pers	<i>nombre</i>	
p_overall_pers	<i>nombre</i>	
r_persistence	<i>nombre</i>	
r_num_clusters	<i>nombre</i>	
r_eta_auto	<i>booléen</i>	

Propriétés neuralnetnode	Valeurs	Description de la propriété
r_alpha	nombre	
r_eta	nombre	
optimize	Speed Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.
calculate_variable_importance	booléen	Remarque : La propriété <code>sensitivity_analysis</code> utilisée dans les versions précédentes est remplacée par cette propriété. L'ancienne propriété est toujours prise en charge, mais il est recommandé d'utiliser <code>calculate_variable_importance</code> .
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés de neuralnetworknode



Le noeud R. neurones est un modèle simplifié de la manière dont le cerveau humain traite les informations. Le fonctionnement de ce modèle repose sur la simulation d'un grand nombre d'unités de traitement simples interconnectées, qui sont en quelque sorte des versions abstraites de nos neurones. Les réseaux de neurones sont de puissants estimateurs de fonctions qui ne requièrent qu'une connaissance limitée en matière de statistiques ou de mathématiques.

Exemple

```
create neuralnetworknode
# Build Options tab - Objectives panel
set: neuralnetworknode.objective = Standard
# Build Options tab - Stopping Rules panel
set: neuralnetworknode.model_selection = BestSubsets
set: neuralnetworknode.criteria_best_subsets = ASE
# Build Options tab - Ensembles panel
set: neuralnetworknode.combining_rule_categorical = HighestMeanProbability
```

Propriétés de neuralnetworknode	Valeurs	Description de la propriété
targets	[champ1 ... champN]	Spécifie des champs cible.
inputs	[champ1 ... champN]	Champs prédicteurs utilisés par le modèle.
splits	[field1 ... fieldN]	Indique le champ ou les champs à utiliser pour la modélisation découpée.
use_partition	Booléen	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
continue	Booléen	Poursuivre le modèle d'apprentissage existant.

Propriétés de neuralnetworknode	Valeurs	Description de la propriété
objective	Standard Bagging Boosting psm	psm est destinée aux ensembles de données très volumineux et nécessite une connexion à un serveur.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	Booléen	
first_layer_units	nombre	
second_layer_units	nombre	
use_max_time	Booléen	
max_time	nombre	
use_max_cycles	Booléen	
max_cycles	nombre	
use_min_accuracy	Booléen	
min_accuracy	nombre	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	nombre	
overfit_prevention_pct	nombre	
use_random_seed	Booléen	
random_seed	nombre	
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	Booléen	
custom_model_name	chaîne	
confidence	onProbability onIncrease	
score_category_probabilities	Booléen	
max_categories	nombre	
score_propensity	Booléen	
use_custom_name	Booléen	
custom_name	chaîne	
tooltip	chaîne	
keywords	chaîne	
annotation	chaîne	

Propriétés de questnode



Le noeud QUEST est une méthode de classification supervisée binaire permettant de créer des arbres décision, développée pour réduire le temps de traitement nécessaire aux analyses C&R Tree importantes, tout en limitant la tendance, observée parmi les méthodes d'arbre de classification, à favoriser les entrées autorisant un nombre supérieur de divisions. Les champs d'entrée peuvent être des intervalles numériques (continues) mais les champs cible doivent être catégoriels. Toutes les divisions sont binaires. [Pour plus d'informations, reportez-vous à la section Noeud QUEST dans le chapitre 6 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create questnode
set :questnode.custom_fields = True
set :questnode.target = Drug
set :questnode.inputs = [Age Na K Cholesterol BP]
set :questnode.model_output_type = InteractiveBuilder
set :questnode.use_tree_directives = True
set :questnode.mode = Expert
set :questnode.max_surrogates = 5
set :questnode.split_alpha = 0.03
set :questnode.use_percentage = False
set :questnode.min_parent_records_abs = 40
set :questnode.min_child_records_abs = 30
set :questnode.prune_tree = True
set :questnode.use_std_err = True
set :questnode.std_err_multiplier = 3
set :questnode.priors = Custom
set :questnode.custom_priors = [{drugA 0.3}{drugB 0.4}]
set :questnode.adjust_priors = true
```

Propriétés questnode	Valeurs	Description de la propriété
target	<i>champ</i>	Les modèles QUEST requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ d'effectifs peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue_training_existing_model	<i>booléen</i>	
objective	Standard Boosting Bagging psm	psm est destinée aux ensembles de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>booléen</i>	
tree_directives	<i>chaîne</i>	
use_max_depth	Default Custom	

Propriétés questnode	Valeurs	Description de la propriété
max_depth	<i>entier</i>	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom
prune_tree	<i>booléen</i>	Elagage de l'arbre pour éviter le surajustement.
use_std_err	<i>booléen</i>	Utiliser la différence maximale pour le risque (dans Erreurs standard).
std_err_multiplier	<i>nombre</i>	Différence maximale.
max_surrogates	<i>nombre</i>	Nombre maximal de substitutions.
use_percentage	<i>booléen</i>	
min_parent_records_pc	<i>nombre</i>	
min_child_records_pc	<i>nombre</i>	
min_parent_records_abs	<i>nombre</i>	
min_child_records_abs	<i>nombre</i>	
use_costs	<i>booléen</i>	
costs	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , où les arguments entre <code>{}</code> sont les coûts prévus réels.
priors	Data Equal Custom	
custom_priors	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>set :cartnode. custom_priors = [{ drugA 0.3 } { drugB 0.6 }]</code>
adjust_priors	<i>booléen</i>	
trails	<i>nombre</i>	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	<i>booléen</i>	Appliquer le boosting aux ensembles de données très volumineux.
split_alpha	<i>nombre</i>	Niveau de signification pour la division.
train_pct	<i>nombre</i>	Ensemble de prévention de surajustement.
set_random_seed	<i>booléen</i>	Dupliquer l'option des résultats.
seed	<i>nombre</i>	
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés de regressionnode



La régression linéaire est une technique statistique couramment utilisée dans le domaine de la synthèse de données et de la prévision. Cette technique établit une ligne droite ou une surface afin de réduire les écarts entre les valeurs de sortie prévues et observées.

Remarque : le noeud Régression sera remplacé par le noeud Linéaire dans une prochaine version. Nous recommandons d'ores et déjà d'utiliser les modèles linéaires pour la régression linéaire.

Exemple

```
create regressionnode
#"Fields" tab
set :regressionnode.custom_fields = True
set :regressionnode.target = 'Age'
set :regressionnode.inputs = ['Na' 'K']
set :regressionnode.partition = Test
set :regressionnode.use_weight = True
set :regressionnode.weight_field = 'Drug'
#"Model" tab
set :regressionnode.use_model_name = False
set :regressionnode.model_name = "Regression Age"
set :regressionnode.use_partitioned_data = True
set :regressionnode.method = Stepwise
set :regressionnode.include_constant = False
#"Expert" tab
set :regressionnode.mode = Expert
set :regressionnode.complete_records = False
set :regressionnode.tolerance = "1.0E-3"
#"Stepping..." section
set :regressionnode.stepping_method = Probability
set :regressionnode.probability_entry = 0.77
set :regressionnode.probability_removal = 0.88
set :regressionnode.F_value_entry = 7.0
set :regressionnode.F_value_removal = 8.0
#"Output..." section
set :regressionnode.model_fit = True
set :regressionnode.r_squared_change = True
set :regressionnode.selection_criteria = True
set :regressionnode.descriptives = True
set :regressionnode.p_correlations = True
set :regressionnode.collinearity_diagnostics = True
set :regressionnode.confidence_interval = True
set :regressionnode.covariance_matrix = True
```

```
set :regressionnode.durbin_watson = True
```

Propriétés regressionnode	Valeurs	Description de la propriété
target	champ	Les modèles Régression requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
method	Enter Stepwise Backwards Forwards	
include_constant	booléen	
use_weight	booléen	
weight_field	champ	
mode	Simple Expert	
complete_records	booléen	
tolerance	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	Pour les arguments, utilisez des guillemets doubles.
stepping_method	useP useF	useP : utiliser la probabilité de F useF : utiliser la valeur F
probability_entry	nombre	
probability_removal	nombre	
F_value_entry	nombre	
F_value_removal	nombre	
selection_criteria	booléen	
confidence_interval	booléen	
covariance_matrix	booléen	
collinearity_diagnostics	booléen	
regression_coefficients	booléen	
exclude_fields	booléen	
durbin_watson	booléen	
model_fit	booléen	
r_squared_change	booléen	
p_correlations	booléen	
descriptives	booléen	
calculate_variable_importance	booléen	

Propriétés de sequencenode



Le noeud Séquence recherche des règles d'association dans des données dotées d'une dimension temporelle. Une séquence est une liste de jeux d'éléments ayant tendance à survenir dans un ordre prévisible. Par exemple, un client qui achète un rasoir et une lotion après-rasage achètera vraisemblablement de la crème à raser. Le noeud Séquence est basé sur l'algorithme de règles d'association CARMA, qui utilise une méthode efficace de double lecture pour rechercher des séquences. [Pour plus d'informations, reportez-vous à la section Noeud Séquence dans le chapitre 12 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create sequencenode
connect :databasenode to :sequencenode
# "Fields" tab
set :sequencenode.id_field = 'Age'
set :sequencenode.contiguous = True
set :sequencenode.use_time_field = True
set :sequencenode.time_field = 'Date1'
set :sequencenode.content_fields = ['Drug' 'BP']
set :sequencenode.partition = Test
# "Model" tab
set :sequencenode.use_model_name = True
set :sequencenode.model_name = "Sequence_test"
set :sequencenode.use_partitioned_data = False
set :sequencenode.min_supp = 15.0
set :sequencenode.min_conf = 14.0
set :sequencenode.max_size = 7
set :sequencenode.max_predictions = 5
# "Expert" tab
set :sequencenode.mode = Expert
set :sequencenode.use_max_duration = True
set :sequencenode.max_duration = 3.0
set :sequencenode.use_pruning = True
set :sequencenode.pruning_value = 4.0
set :sequencenode.set_mem_sequences = True
set :sequencenode.mem_sequences = 5.0
set :sequencenode.use_gaps = True
set :sequencenode.min_item_gap = 20.0
set :sequencenode.max_item_gap = 30.0
```

Propriétés sequencenode	Valeurs	Description de la propriété
id_field	champ	Pour créer un modèle Séquence, vous devez renseigner un champ ID, un champ de temps facultatif et au moins un champ d'analyse. Les champs de pondération et d'effectifs ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
time_field	champ	

Propriétés sequencenode	Valeurs	Description de la propriété
use_time_field	booléen	
content_fields	[champ1 ... champn]	
contiguous	booléen	
min_supp	nombre	
min_conf	nombre	
max_size	nombre	
max_predictions	nombre	
mode	Simple Expert	
use_max_duration	booléen	
max_duration	nombre	
use_gaps	booléen	
min_item_gap	nombre	
max_item_gap	nombre	
use_pruning	booléen	
pruning_value	nombre	
set_mem_sequences	booléen	
mem_sequences	entier	

Propriétés de slrmnode



Le noeud Modèle de réponse en auto-apprentissage (SLRM) vous permet de créer un modèle dans lequel une nouvelle observation unique, ou un petit nombre de nouvelles observations, peuvent être utilisés pour réestimer un modèle sans qu'un recyclage de toutes les données soit nécessaire. [Pour plus d'informations, reportez-vous à la section Noeud MRRA dans le chapitre 14 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create slrmnode
set :slrmnode.target = Offer
set :slrmnode.target_response = Response
set :slrmnode.inputs = ['Cust_ID' 'Age' 'Ave_Bal']
```

Propriétés slrmnode	Valeurs	Description de la propriété
target	champ	Le champ cible doit être nominal ou booléen. Un champ d'effectifs peut aussi être spécifié. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
target_response	champ	Type doit être booléen.
continue_training_existing_model	booléen	

Propriétés slrmnode	Valeurs	Description de la propriété
target_field_values	booléen	Tout utiliser : Utiliser toutes les valeurs source. Spécifier : Sélectionner les valeurs nécessaires.
target_field_values_specify	[champ1 ... champN]	
include_model_assessment	booléen	
model_assessment_random_seed	nombre	Doit être un nombre réel.
model_assessment_sample_size	nombre	Doit être un nombre réel.
model_assessment_iterations	nombre	Nombre d'itérations.
display_model_evaluation	booléen	
max_predictions	nombre	
randomization	nombre	
scoring_random_seed	nombre	
sort	Ascending Descending	Indique si les premières offres affichées sont celles dont les scores sont les plus élevés ou les moins élevés.
model_reliability	booléen	
calculate_variable_importance	booléen	

Propriétés statisticsmodelnode



Le noeud Modèle Statistics vous permet d'analyser et de travailler avec vos données en exécutant des procédures IBM® SPSS® Statistics qui produisent un PMML. Ce noeud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Noeud Modèle Statistics dans le chapitre 8 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Les propriétés de ce noeud sont décrites dans [Propriétés statisticsmodelnode sur p. 317.](#)

Propriétés de svmnode



Le noeud Support Vector Machine (SVM) vous permet de classer les données dans l'un de deux groupes sans sur-ajustement. SVM fonctionne bien avec les grands ensembles de données, comme ceux qui disposent d'un très grand nombre de champs d'entrée. [Pour plus d'informations, reportez-vous à la section Noeud SVM dans le chapitre 15 dans Noeuds de modélisation de IBM SPSS Modeler 15.](#)

Exemple

```
create svmnode
# Expert tab
set :svmnode.mode=Expert
set :svmnode.all_probabilities=True
set :svmnode.kernel=Polynomial
```



```
set :svmnode.gamma=1.5
```

Propriétés svmnode	Valeurs	Description de la propriété
all_probabilities	<i>booléen</i>	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (par défaut) 1.0E-4 1.0E-5 1.0E-6	Détermine le moment de l'arrêt de l'algorithme d'optimisation.
regularization	<i>nombre</i>	Egalement appelé Paramètre C
precision	<i>nombre</i>	Option uniquement utilisée si le niveau de mesure du champ cible est Continuous.
kernel	RBF (par défaut) Polynomial Sigmoid Linear	Type de fonction de noyau utilisée pour la transformation.
rbf_gamma	<i>nombre</i>	Utilisé uniquement si kernel est RBF.
gamma	<i>nombre</i>	Utilisé uniquement si kernel est Polynomial ou Sigmoid.
bias	<i>nombre</i>	
degree	<i>nombre</i>	Utilisé uniquement si kernel est Polynomial.
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés de timeseriesnode



Le noeud Séries temporelles estime les modèles de lissage exponentiel, d'ARIMA (Autoregressive Integrated Moving Average) univariable et d'ARIMA multivariable (ou fonction de transfert) pour les données de séries temporelles et génère des prévisions d'une performance future. Un noeud Séries temporelles doit toujours être précédé d'un noeud Intervalles de temps. [Pour plus d'informations, reportez-vous à la section Noeud de modélisation Séries temporelles dans le chapitre 13 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create timeseriesnode
set :timeseriesnode.method = Exsmooth
set :timeseriesnode.exsmooth_model_type = HoltsLinearTrend
set :timeseriesnode.exsmooth_transformation_type = None
```

Propriétés timeseriesnode	Valeurs	Description de la propriété
targets	<i>champ</i>	Le noeud Séries temporelles prévoit une ou plusieurs cibles, utilisant éventuellement un ou plusieurs champs d'entrée en tant que variables indépendantes. Les champs d'effectifs et de pondération ne sont pas utilisés. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
continue	<i>booléen</i>	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	<i>booléen</i>	
consider_seasonal	<i>booléen</i>	
detect_outliers	<i>booléen</i>	
expert_outlier_additive	<i>booléen</i>	
expert_outlier_level_shift	<i>booléen</i>	
expert_outlier_innovational	<i>booléen</i>	
expert_outlier_level_shift	<i>booléen</i>	
expert_outlier_transient	<i>booléen</i>	
expert_outlier_seasonal_additive	<i>booléen</i>	
expert_outlier_local_trend	<i>booléen</i>	
expert_outlier_additive_patch	<i>booléen</i>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	<i>entier</i>	
arima_d	<i>entier</i>	
arima_q	<i>entier</i>	
arima_sp	<i>entier</i>	
arima_sd	<i>entier</i>	
arima_sq	<i>entier</i>	

Propriétés timeseriesnode	Valeurs	Description de la propriété
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	booléen	
tf_arima_p.fieldname	entier	Pour les fonctions de transfert.
tf_arima_d.fieldname	entier	Pour les fonctions de transfert.
tf_arima_q.fieldname	entier	Pour les fonctions de transfert.
tf_arima_sp.fieldname	entier	Pour les fonctions de transfert.
tf_arima_sd.fieldname	entier	Pour les fonctions de transfert.
tf_arima_sq.fieldname	entier	Pour les fonctions de transfert.
tf_arima_delay.fieldname	entier	Pour les fonctions de transfert.
tf_arima_transformation_type.fieldname	None SquareRoot NaturalLog	Pour les fonctions de transfert.
arima_detect_outlier_mode	None Automatic	
arima_outlier_additive	booléen	
arima_outlier_level_shift	booléen	
arima_outlier_innovational	booléen	
arima_outlier_transient	booléen	
arima_outlier_seasonal_additive	booléen	
arima_outlier_local_trend	booléen	
arima_outlier_additive_patch	booléen	
conf_limit_pct	réel	
max_lags	entier	
events	champs	
scoring_model_only	booléen	Utilisation pour les modèles comportant un grand nombre (des dizaines de milliers) de séries temporelles.

Propriétés de twostepnode



Le noeud TwoStep utilise une méthode de classification non supervisée en deux étapes. La première étape consiste en une exploration des données visant à compresser les données d'entrée brutes en sous-classes plus faciles à manipuler. Au cours de la seconde étape, l'utilisation d'une méthode de classification hiérarchique permet de fusionner progressivement les sous-classes en classes de plus en plus importantes. La technique TwoStep a l'avantage d'évaluer automatiquement le nombre de classes optimal pour les données d'apprentissage. Il peut prendre en charge de manière efficace des types de champ mixtes et des ensembles de données volumineux. [Pour plus d'informations, reportez-vous à la section Noeud Classification TwoStep dans le chapitre 11 dans *Noeuds de modélisation de IBM SPSS Modeler 15*.](#)

Exemple

```
create twostep
set :twostep.custom_fields = True
set :twostep.inputs = ['Age' 'K' 'Na' 'BP']
set :twostep.partition = Test
set :twostep.use_model_name = False
set :twostep.model_name = "TwoStep_Drug"
set :twostep.use_partitioned_data = True
set :twostep.exclude_outliers = True
set :twostep.cluster_label = "String"
set :twostep.label_prefix = "TwoStep_"
set :twostep.cluster_num_auto = False
set :twostep.max_num_clusters = 9
set :twostep.min_num_clusters = 3
set :twostep.num_clusters = 7
```

Propriétés twostepnode	Valeurs	Description de la propriété
inputs	[<i>champ1 ... champN</i>]	Les modèles TwoStep utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et d'effectifs ne sont pas reconnus. Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds de modélisation sur p. 197.
standardize	booléen	
exclude_outliers	booléen	
percentage	nombre	
cluster_num_auto	booléen	
min_num_clusters	nombre	
max_num_clusters	nombre	
num_clusters	nombre	
cluster_label	String Number	
label_prefix	chaîne	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

Propriétés du noeud de nugget de modèle

Les noeuds de nugget de modèle partagent les mêmes propriétés que les autres noeuds. [Pour plus d'informations, reportez-vous à la section Propriétés communes des noeuds dans le chapitre 9 sur p. 118.](#)

Propriétés de *applyanomalydetectionnode*

Les noeuds de modélisation Détection des anomalies peuvent être utilisés pour générer un nugget de modèle Détection des anomalies. Le nom de génération de script de ce nugget de modèle est *applyanomalydetectionnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, [voir Propriétés de *anomalydetectionnode* dans le chapitre 16 sur p. 198.](#)

Propriétés de <i>applyanomalydetectionnode</i>	Valeurs	Description de la propriété
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Détermine les sorties créées pour le scoring.
<code>num_fields</code>	<i>entier</i>	Champs à signaler.
<code>discard_records</code>	<i>Booléen</i>	Indique si les enregistrements sont supprimés de la sortie.
<code>discard_anomalous_records</code>	<i>Booléen</i>	Indique s'il faut supprimer les enregistrements irréguliers ou <i>réguliers</i> . Cette propriété est désactivée par défaut, ce qui signifie que les enregistrements <i>réguliers</i> sont supprimés. Si elle est <i>on</i> , les enregistrements irréguliers sont supprimés. Cette propriété n'est activée que si la propriété <code>discard_records</code> l'est également.

Propriétés de *applyapriorinode*

Les noeuds de modélisation Apriori peuvent être utilisés pour générer un nugget de modèle Apriori. Le nom de génération de script de ce nugget de modèle est *applyapriorinode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, [voir Propriétés de *apriorinode* dans le chapitre 16 sur p. 199.](#)

Propriétés de <i>applyapriorinode</i>	Valeurs	Description de la propriété
<code>max_predictions</code>	<i>nombre (entier)</i>	
<code>ignore_unmatached</code>	<i>Booléen</i>	
<code>allow_repeats</code>	<i>Booléen</i>	

Propriétés de <i>applypriorinode</i>	Valeurs	Description de la propriété
check_basket	NoPredictions Predictions NoCheck	
criterion	Confidence Support RuleSupport Lift Deployability	

Propriétés de *applyautoclassifiernode*

Les noeuds de modélisation de classificateur automatique peuvent être utilisés pour générer un nugget de modèle de classificateur automatique. Le nom de script de ce nugget d'un modèle est *applyautoclassifiernode*. Pour de plus amples informations sur la génération de scripts du noeud de modèles lui-même, [reportez-vous à Propriétés autoclassifiernode dans le chapitre 16 sur p. 201.](#)

Propriétés de <i>applyautoclassifiernode</i>	Valeurs	Description de la propriété
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ booléen.
flag_voting_tie_selection	Random HighestConfidence RawPropensity	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ booléen.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ d'ensemble.
set_voting_tie_selection	Random HighestConfidence	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.

Propriétés de *applyautoclusternode*

Les noeuds de modélisation de classification automatique peuvent être utilisés pour générer un nugget de modèle de classification automatique. Le nom de génération de script de ce nugget de modèle est *applydecisionlistnode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, [voir Propriétés du noeud de classification automatique dans le chapitre 16 sur p. 203.](#)

Propriétés de *applyautonumericnode*

Les noeuds de modélisation de numérisation automatique peuvent être utilisés pour générer un nugget de modèle de numérisation automatique. Le nom de script de ce nugget d'un modèle est *applyautonumericnode*. Pour de plus amples informations sur la génération de scripts du noeud de modélisation lui-même, [reportez-vous à Propriétés de autonumericnode dans le chapitre 16 sur p. 205](#).

Propriétés de <i>applyautonumericnode</i>	Valeurs	Description de la propriété
calculate_standard_error	Booléen	

Propriétés de *applybayesnetnode*

Les noeuds de modélisation Réseau Bayésien peuvent être utilisés pour générer un nugget de modèle Réseau Bayésien. Le nom de génération de script de ce nugget de modèle est *applybayesnetnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, [voir Propriétés de bayesnetnode dans le chapitre 16 sur p. 206](#).

Propriétés de <i>applybayesnetnode</i>	Valeurs	Description de la propriété
all_probabilities	Booléen	
raw_propensity	Booléen	
adjusted_propensity	Booléen	
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applyc50node*

Les noeuds de modélisation C5.0 peuvent être utilisés pour générer un nugget de modèle C5.0. Le nom de génération de script de ce nugget de modèle est *applyc50node*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, [voir Propriétés de c50node dans le chapitre 16 sur p. 208](#).

Propriétés de <i>applyc50node</i>	Valeurs	Description de la propriété
sql_generate	Never NoMissingValues	Permet de définir les options de génération SQL au cours de l'exécution des ensembles de règles.
calculate_conf	Booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applycarmanode*

Les noeuds de modélisation CARMA peuvent être utilisés pour générer un nugget de modèle CARMA. Le nom de génération de script de ce nugget de modèle est *applycarmanode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de carmanode dans le chapitre 16 sur p. 209](#).

Propriétés de *applycartnode*

Des noeuds de modélisation d'arbre C&R peuvent être utilisés pour générer un nugget de modèle d'arbre C&R. Le nom de génération de script de ce nugget de modèle est *applycartnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de cartnode dans le chapitre 16 sur p. 210](#).

Propriétés de <i>applycartnode</i>	Valeurs	Description de la propriété
sql_generate	Never MissingValues NoMissingValues	Permet de définir les options de génération SQL au cours de l'exécution des ensembles de règles.
calculate_conf	Booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
display_rule_id	Booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applychaidnode*

Les noeuds de modélisation CHAID peuvent être utilisés pour générer un nugget de modèle CHAID. Le nom de génération de script de ce nugget de modèle est *applychaidnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de chaidnode dans le chapitre 16 sur p. 213](#).

Propriétés de <i>applychaidnode</i>	Valeurs	Description de la propriété
sql_generate	Never MissingValues	
calculate_conf	Booléen	
display_rule_id	Booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applycoxregnode*

Les noeuds de modélisation de Cox peuvent être utilisés pour générer un nugget de modèle de Cox. Le nom de génération de script de ce nugget de modèle est *applycoxregnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de coxregnode dans le chapitre 16 sur p. 215](#).

Propriétés de <i>applycoxregnode</i>	Valeurs	Description de la propriété
future_time_as	Intervals Fields	
time_interval	<i>nombre</i>	
num_future_times	<i>entier</i>	
time_field	<i>champ</i>	
past_survival_time	<i>champ</i>	
all_probabilities	<i>Booléen</i>	
cumulative_hazard	<i>Booléen</i>	

Propriétés de *applydecisionlistnode*

Les noeuds de modélisation Liste de décision peuvent être utilisés pour générer un nugget de modèle Liste de décision. Le nom de génération de script de ce nugget de modèle est *applydecisionlistnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de decisionlistnode dans le chapitre 16 sur p. 217](#).

Propriétés de <i>applydecisionlistnode</i>	Valeurs	Description de la propriété
enable_sql_generation	<i>Booléen</i>	Lorsque la valeur est true (vrai), IBM® SPSS® Modeler essaie de répercuter le modèle Liste de décision dans SQL.
calculate_raw_propensities	<i>Booléen</i>	
calculate_adjusted_propensities	<i>Booléen</i>	

Propriétés de *applydiscriminantnode*

Les noeuds de modélisation Discriminant peuvent être utilisés pour générer un nugget de modèle Discriminant. Le nom de génération de script de ce nugget de modèle est *applydiscriminantnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de discriminantnode dans le chapitre 16 sur p. 218](#).

Propriétés de <i>applydiscriminantnode</i>	Valeurs	Description de la propriété
calculate_raw_propensities	<i>Booléen</i>	
calculate_adjusted_propensities	<i>Booléen</i>	

Propriétés de *applyfactornode*

Les noeuds de modélisation APC/Facteur peuvent être utilisés pour générer un nugget de modèle APC/Facteur. Le nom de génération de script de ce nugget de modèle est *applyfactornode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de factornode dans le chapitre 16 sur p. 220](#).

Propriétés de *applyfeatureselectionnode*

Les noeuds de modélisation Sélection de fonction peuvent être utilisés pour générer un nugget de modèle Sélection de fonction. Le nom de génération de script de ce nugget de modèle est *applyfeatureselectionnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de featureselectionnode dans le chapitre 16 sur p. 222](#).

Propriétés de <i>applyfeatureselectionnode</i>	Valeurs	Description de la propriété
selected_ranked_fields		Indique les champs classés sélectionnés dans le navigateur de modèle.
selected_screened_fields		Indique les champs filtrés sélectionnés dans le navigateur de modèle.

Propriétés de *applygeneralizedlinearnode*

Les noeuds de modélisation linéaire généralisée (genlin) peuvent être utilisés pour générer un nugget de modèle linéaire généralisé. Le nom de génération de script de ce nugget de modèle est *applygeneralizedlinearnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de genlinnode dans le chapitre 16 sur p. 224](#).

Propriétés de <i>applygeneralizedlinearnode</i>	Valeurs	Description de la propriété
calculate_raw_propensities	<i>Booléen</i>	
calculate_adjusted_propensities	<i>Booléen</i>	

Propriétés de *applykmeansnode*

Les noeuds de modélisation K-Means peuvent être utilisés pour générer un nugget de modèle K-Means. Le nom de génération de script de ce nugget de modèle est *applykmeansnode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de kmeansnode dans le chapitre 16 sur p. 230](#).

Propriétés *applyknnnode*

Les noeuds de modélisation KNN peuvent être utilisés pour générer un nugget de modèle KNN. Le nom de génération de script de ce nugget de modèle est *applyknnnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés knnnode dans le chapitre 16 sur p. 231](#).

Propriétés <i>applyknnnode</i>	Valeurs	Description de la propriété
<code>all_probabilities</code>	<i>Booléen</i>	
<code>save_distances</code>	<i>Booléen</i>	

Propriétés de *applykohonennode*

Les noeuds de modélisation Kohonen peuvent être utilisés pour générer un nugget de modèle Kohonen. Le nom de génération de script de ce nugget de modèle est *applykohonennode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de c50node dans le chapitre 16 sur p. 208](#).

Propriétés de *applylinearnode*

Les noeuds de modélisation linéaire peuvent être utilisés pour générer un nugget de modèle linéaire. Le nom de génération de script de ce nugget de modèle est *applylinearnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de linearnode dans le chapitre 16 sur p. 234](#).

Propriétés <i>linear</i>	Valeurs	Description de la propriété
<code>use_custom_name</code>	<i>Booléen</i>	
<code>custom_name</code>	<i>chaîne</i>	
<code>enable_sql_generation</code>	<i>Booléen</i>	

Propriétés de *applylogregnode*

Les noeuds de modélisation Régression logistique peuvent être utilisés pour générer un nugget de modèle Régression logistique. Le nom de génération de script de ce nugget de modèle est *applylogregnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de logregnode dans le chapitre 16 sur p. 235](#).

Propriétés de <i>applylogregnode</i>	Valeurs	Description de la propriété
<code>calculate_raw_propensities</code>	<i>Booléen</i>	

Propriétés de *applyneuralnetnode*

Les noeuds de modélisation Réseau de neurones peuvent être utilisés pour générer un nugget de modèle Réseau de neurones. Le nom de génération de script de ce nugget de modèle est *applyneuralnetnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de *neuralnetnode* dans le chapitre 16 sur p. 240.](#)

Avertissement : Une nouvelle version du nugget Réseau de neurones, avec des fonctions améliorées, est disponible dans cette version et est décrite dans la section suivante (*applyneuralnetwork*). Bien que la version précédente soit encore disponible, nous vous conseillons de procéder à la mise à jour de vos scripts afin d'utiliser la nouvelle version. Les détails de la version précédente sont préservés ici à titre de référence, mais la prise en charge sera supprimée dans une version ultérieure.

Propriétés de <i>applyneuralnetnode</i>	Valeurs	Description de la propriété
calculate_conf	Booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
enable_sql_generation	Booléen	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applyneuralnetworknode*

Les noeuds de modélisation Réseau de neurones peuvent être utilisés pour générer un nugget de modèle Réseau de neurones. Le nom de génération de script de ce nugget de modèle est *applyneuralnetworknode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de *neuralnetworknode* dans le chapitre 16 sur p. 242.](#)

Propriétés de <i>applyneuralnetworknode</i>	Valeurs	Description de la propriété
use_custom_name	Booléen	
custom_name	chaîne	
confidence	onProbability onIncrease	
score_category_probabilities	Booléen	
max_categories	nombre	
score_propensity	Booléen	

Propriétés de *applyquestnode*

Les noeuds de modélisation QUEST peuvent être utilisés pour générer un nugget de modèle QUEST. Le nom de génération de script de ce nugget de modèle est *applyquestnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de questnode dans le chapitre 16 sur p. 244](#).

Propriétés de <i>applyquestnode</i>	Valeurs	Description de la propriété
sql_generate	Never MissingValues NoMissingValues	
calculate_conf	Booléen	
display_rule_id	Booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	Booléen	
calculate_adjusted_propensities	Booléen	

Propriétés de *applyregressionnode*

Les noeuds de modélisation Régression linéaire peuvent être utilisés pour générer un nugget de modèle Régression linéaire. Le nom de génération de script de ce nugget de modèle est *applyregressionnode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de regressionnode dans le chapitre 16 sur p. 246](#).

Propriétés de *applyselflearningnode*

Les noeuds de modélisation Réponse en auto-apprentissage (SLRM) peuvent être utilisés pour générer un nugget de modèle SLRM. Le nom de génération de script de ce nugget de modèle est *applyselflearningnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de slrmnode dans le chapitre 16 sur p. 249](#).

Propriétés de <i>applyselflearningnode</i>	Valeurs	Description de la propriété
max_predictions	nombre	
randomization	nombre	
scoring_random_seed	nombre	
sort	ascending descending	Indique si les premières offres affichées sont celles dont les scores sont les plus élevés ou les moins élevés.
model_reliability	Booléen	Option Prendre en compte la fiabilité du modèle dans l'onglet Paramètres.

Propriétés de *applysequencenode*

Les noeuds de modélisation Séquence peuvent être utilisés pour générer un nugget de modèle Séquence. Le nom de génération de script de ce nugget de modèle est *applysequencenode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de sequencenode dans le chapitre 16 sur p. 248](#).

Propriétés de *applysvmnode*

Les noeuds de modélisation SVM peuvent être utilisés pour générer un nugget de modèle SVM. Le nom de génération de script de ce nugget de modèle est *applysvmnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de svmnode dans le chapitre 16 sur p. 250](#).

Propriétés de <i>applysvmnode</i>	Valeurs	Description de la propriété
<code>all_probabilities</code>	<i>Booléen</i>	
<code>calculate_raw_propensities</code>	<i>Booléen</i>	
<code>calculate_adjusted_propensities</code>	<i>Booléen</i>	

Propriétés de *applytimeseriesnode*

Les noeuds de modélisation Séries temporelles peuvent être utilisés pour générer un nugget de modèle Séries temporelles. Le nom de génération de script de ce nugget de modèle est *applytimeseriesnode*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de timeseriesnode dans le chapitre 16 sur p. 251](#).

Propriétés de <i>applytimeseriesnode</i>	Valeurs	Description de la propriété
<code>calculate_conf</code>	<i>Booléen</i>	
<code>calculate_residuals</code>	<i>Booléen</i>	

Propriétés de *applytwostepnode*

Les noeuds de modélisation TwoStep peuvent être utilisés pour générer un nugget de modèle TwoStep. Le nom de génération de script de ce nugget de modèle est *applytwostepnode*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir [Propriétés de twostepnode dans le chapitre 16 sur p. 254](#).

Propriétés du nœud de modélisation SGBD

IBM® SPSS® Modeler prend en charge l'intégration des outils de Data mining et de modélisation disponibles auprès des fournisseurs de base de données, notamment Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM® DB2® InfoSphere Warehouse et IBM® Netezza® Analytics. [Pour plus d'informations, reportez-vous à la section Modélisation de base de données - Présentation dans le chapitre 2 dans *Guide d'exploration de base de données de IBM SPSS Modeler 15*](#). Vous pouvez créer et déterminer le score des modèles à l'aide d'algorithmes natifs de base de données, qui proviennent tous de l'application SPSS Modeler. Les modèles de base de données peuvent également être créés et manipulés via des scripts à l'aide des propriétés décrites dans cette section.

Par exemple, l'extrait de script suivant illustre la création d'un modèle d'arbre décision Microsoft via l'interface de génération de scripts de SPSS Modeler :

```
create mstreenode
rename :mstreenode as msbuilder
set msbuilder.analysis_server_name = 'localhost'
set msbuilder.analysis_database_name = 'TESTDB'
set msbuilder.mode = 'Expert'
set msbuilder.datasource = 'LocalServer'
set msbuilder.target = 'Drug'
set msbuilder.inputs = ['Age' 'Sex']
set msbuilder.unique_field = 'IDX'
set msbuilder.custom_fields = true
set msbuilder.model_name = 'MSDRUG'

connect :typenode to msbuilder
execute msbuilder

insert model MSDRUG connected between :typenode and :tablenode
set MSDRUG.sql_generate = true
execute :tablenode
```

Propriétés du nœud pour la modélisation Microsoft

Propriétés des nœuds de modélisation Microsoft

Propriétés communes

Les propriétés suivantes sont communes aux nœuds de modélisation de base de données Microsoft.

Propriétés communes des nœuds Microsoft	Valeurs	Description de la propriété
analysis_database_name	chaîne	Nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte Analysis Services.
use_transactional_data	booléen	Spécifie si les données d'entrée sont au format tabulaire ou transactionnel.
inputs	[champ champ champ]	Champs d'entrée pour les données tabulaires.
target	champ	Champ prédit (ne s'applique pas au nœuds Classification non supervisée MS ou Classification de séquences).
unique_field	champ	Champ-clé.
msas_parameters	structurées	Paramètres d'algorithme. Pour plus d'informations, reportez-vous à la section Paramètres d'algorithme sur p. 267.
with_drillthrough	booléen	Option Avec extraction

Arbre décision MS

Aucune propriété particulière n'est définie pour les nœuds du type `mstreenode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Classification non supervisée MS

Aucune propriété particulière n'est définie pour les nœuds du type `msclusternode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Règles d'association MS

Les propriétés spécifiques suivantes sont disponibles pour les nœuds du type `msassocnode` :

Propriétés <code>msassocnode</code>	Valeurs	Description de la propriété
id_field	champ	Identifie chaque transaction dans les données.
trans_inputs	[champ champ champ]	Champs d'entrée pour les données transactionnelles.
transactional_target	champ	Champ variable indépendante (données transactionnelles).

MS Naive Bayes

Aucune propriété particulière n'est définie pour les noeuds du type `msbayesnode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Régression linéaire MS

Aucune propriété particulière n'est définie pour les noeuds du type `msregressionnode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Réseau neuronal MS

Aucune propriété particulière n'est définie pour les noeuds du type `msneuralnetworknode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Régression logistique MS

Aucune propriété particulière n'est définie pour les noeuds du type `mslogisticnode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Séries temporelles MS

Aucune propriété particulière n'est définie pour les noeuds du type `mstimeseriesnode`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

MS Sequence Clustering

Les propriétés spécifiques suivantes sont disponibles pour les noeuds du type `mssequenceclusternode` :

Propriétés <code>mssequenceclusternode</code>	Valeurs	Description de la propriété
<code>id_field</code>	<i>champ</i>	Identifie chaque transaction dans les données.
<code>input_fields</code>	<i>[champ champ champ]</i>	Champs d'entrée pour les données transactionnelles.
<code>sequence_field</code>	<i>champ</i>	Identificateur de séquence.
<code>target_field</code>	<i>champ</i>	Champ variable indépendante (données tabulaires).

Paramètres d'algorithme

Chaque type de modèle de base de données Microsoft a des paramètres spécifiques pouvant être définis à l'aide de la propriété `msas_parameters`, par exemple :

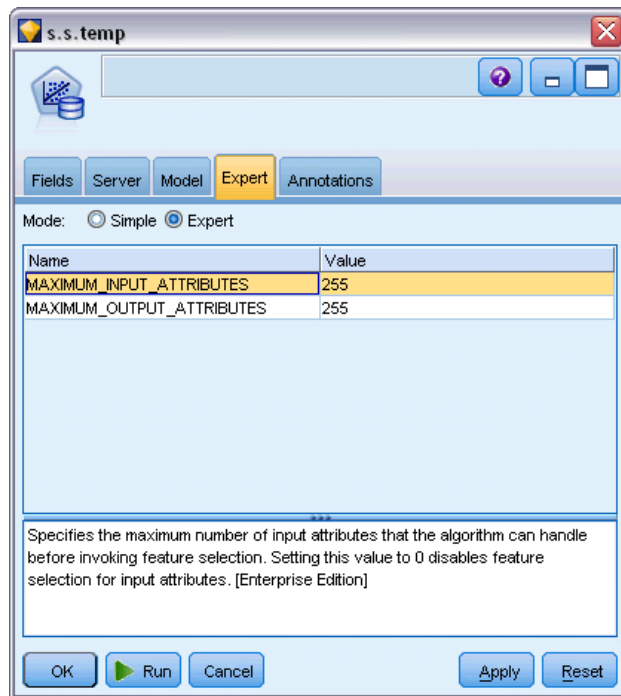
```
set :msregressionnode.msas_parameters =
[{"MAXIMUM_INPUT_ATTRIBUTES" 255},{"MAXIMUM_OUTPUT_ATTRIBUTES" 255}]
```

Ces paramètres sont issus de SQL Server. Pour visualiser les paramètres pertinents de chaque noeud, procédez comme suit :

- ▶ Placez un noeud source de base de données dans l'espace de travail.
- ▶ Ouvrez le noeud source de base de données.
- ▶ Sélectionnez une source valide dans la liste déroulante Source de données.
- ▶ Sélectionnez une table valide dans la liste Nom de la table.
- ▶ Cliquez sur OK pour fermer le noeud source de base de données.
- ▶ Reliez le noeud de modélisation de base de données Microsoft dont vous voulez répertorier les propriétés.
- ▶ Ouvrez le noeud de modélisation de base de données.
- ▶ Cliquez sur l'onglet Expert.

Les propriétés `msas_parameters` disponibles de ce noeud apparaissent.

Figure 18-1
Exemple d'affichage de paramètre d'algorithme



Propriétés du nugget de modèle Microsoft

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des noeuds de modélisation de la base de données Microsoft.

Arbre décision MS

Propriétés applymstreenode	Valeurs	Description
analysis_database_name	chaîne	Ce nœud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.
datasource	chaîne	Nom de la source de données (DSN) ODBC du serveur SQL.
sql_generate	booléen	Active la génération SQL.

Régression linéaire MS

Propriétés applymstreenode	Valeurs	Description
analysis_database_name	chaîne	Ce nœud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Réseau neuronal MS

Propriétés applymstreenode	Valeurs	Description
analysis_database_name	chaîne	Ce nœud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Régression logistique MS

Propriétés applymstreenode	Valeurs	Description
analysis_database_name	chaîne	Ce nœud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Séries temporelles MS

Propriétés applymstreenode	Valeurs	Description
analysis_database_name	chaîne	Ce nœud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Propriétés appliedtime-seriesnode	Valeurs	Description
start_from	new_prediction historical_prediction	Spécifie s'il faut faire des prévisions futures ou des prévisions historiques.
new_step	<i>nombre</i>	Définit la période de commencement des prévisions futures.
historical_step	<i>nombre</i>	Définit la période de commencement des prévisions historiques.
end_step	<i>nombre</i>	Définit la période de fin des prévisions.

MS Sequence Clustering

Propriétés appliedmssequenceclusternode	Valeurs	Description
analysis_database_name	<i>chaîne</i>	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	<i>chaîne</i>	Nom de l'hôte du serveur Analysis.

Propriétés du noeud pour la modélisation Oracle

Propriétés du noeud de modélisation Oracle

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données Oracle.

Propriétés communes des noeuds Oracle	Valeurs	Description de la propriété
target	<i>champ</i>	
inputs	<i>Liste des champs</i>	
partition	<i>champ</i>	Champ utilisé pour partitionner les données en échantillons distincts pour les étapes d'apprentissage, de test et de validation de la création d'un modèle.
datasource		
username		
password		
epassword		
use_model_name	<i>booléen</i>	
model_name	<i>chaîne</i>	Nom personnalisé du nouveau modèle.
use_partitioned_data	<i>booléen</i>	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
unique_field	<i>champ</i>	
auto_data_prep	<i>booléen</i>	Active ou désactive la fonction de préparation de données automatique d'Oracle (Bases de données 11g uniquement).

Propriétés communes des noeuds Oracle	Valeurs	Description de la propriété
costs	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , où les arguments entre <code>{ }</code> sont les coûts prévus réels.
mode	Simple Expert	Certaines propriétés sont ignorées si ce paramètre est défini sur Simple, comme indiqué dans les propriétés de noeud individuelles.
use_prediction_probability	<i>booléen</i>	
prediction_probability	<i>chaîne</i>	
use_prediction_set	<i>booléen</i>	

Oracle Naive Bayes

Les propriétés suivantes sont disponibles pour les noeuds du type oranbnode.

Propriétés oranbnode	Valeurs	Description de la propriété
singleton_threshold	<i>nombre</i>	0.0–1.0.*
pairwise_threshold	<i>nombre</i>	0.0–1.0.*
priors	Data Equal Custom	
custom_priors	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>set :oranbnode.custom_priors = {{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}}</code>

* Propriété ignorée si mode est défini sur Simple.

Oracle Adaptive Bayes

Les propriétés suivantes sont disponibles pour les noeuds du type oraabnnode.

Propriétés oraabnnode	Valeurs	Description de la propriété
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	<i>booléen</i>	*
execution_time_limit	<i>entier</i>	La valeur doit être supérieure à 0.*
max_naive_bayes_predictors	<i>entier</i>	La valeur doit être supérieure à 0.*
max_predictors	<i>entier</i>	La valeur doit être supérieure à 0.*
priors	Data Equal Custom	
custom_priors	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>set :oraabnnode.custom_priors = {{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}}</code>

* Propriété ignorée si mode est défini sur Simple.

Oracle Support Vector Machines

Les propriétés suivantes sont disponibles pour les noeuds du type orasvmnode.

Propriétés orasvmnode	Valeurs	Description de la propriété
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	<i>entier</i>	Noyau gaussien uniquement. La valeur doit être supérieure à 0.*
convergence_tolerance	<i>nombre</i>	La valeur doit être supérieure à 0.*
use_standard_deviation	<i>booléen</i>	Noyau gaussien uniquement.*
standard_deviation	<i>nombre</i>	La valeur doit être supérieure à 0.*
use_epsilon	<i>booléen</i>	Modèles de régression uniquement.*
epsilon	<i>nombre</i>	La valeur doit être supérieure à 0.*
use_complexity_factor	<i>booléen</i>	*
complexity_factor	<i>nombre</i>	*
use_outlier_rate	<i>booléen</i>	Variante à classe unique seulement.*
outlier_rate	<i>nombre</i>	Variante à classe unique seulement. 0.0–1.0.*
weights	Data Equal Custom	
custom_weights	<i>structurées</i>	Propriété structurée utilisant la syntaxe : set :orasvmnode.custom_weights = [{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}]

* Propriété ignorée si mode est défini sur Simple.

Modèles linéaires généralisés d'Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oraglmnode.

Propriétés oraglmnode	Valeurs	Description de la propriété
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWith- Mean UseCompleteRe- cords	
use_row_weights	<i>booléen</i>	*
row_weights_field	<i>champ</i>	*
save_row_diagnostics	<i>booléen</i>	*
row_diagnostics_table	<i>chaîne</i>	*
coefficient_confidence	<i>nombre</i>	*

Propriétés oraglmnode	Valeurs	Description de la propriété
use_reference_category	<i>booléen</i>	*
reference_category	<i>chaîne</i>	*
ridge_regression	Auto Off On	*
parameter_value	<i>nombre</i>	*
vif_for_ridge	<i>booléen</i>	*

* Propriété ignorée si mode est défini sur Simple.

Arbre décision Oracle

Les propriétés suivantes sont disponibles pour les nœuds du type oradecisiontreenode.

Propriétés oradecisiontreenode	Valeurs	Description de la propriété
use_costs	<i>booléen</i>	
impurity_metric	Entropy Gini	
term_max_depth	<i>entier</i>	2–20.*
term_minpct_node	<i>nombre</i>	0.0–10.0.*
term_minpct_split	<i>nombre</i>	0.0–20.0.*
term_minrec_node	<i>entier</i>	La valeur doit être supérieure à 0.*
term_minrec_split	<i>entier</i>	La valeur doit être supérieure à 0.*
display_rule_ids	<i>booléen</i>	*

* Propriété ignorée si mode est défini sur Simple.

O-Cluster Oracle

Les propriétés suivantes sont disponibles pour les nœuds du type oraclusternode.

Propriétés oraclusternode	Valeurs	Description de la propriété
max_num_clusters	<i>entier</i>	La valeur doit être supérieure à 0.
max_buffer	<i>entier</i>	La valeur doit être supérieure à 0.*
sensitivity	<i>nombre</i>	0.0–1.0.*

* Propriété ignorée si mode est défini sur Simple.

KMeans Oracle

Les propriétés suivantes sont disponibles pour les nœuds du type orakmeansnode.

Propriétés orakmeansnode	Valeurs	Description de la propriété
num_clusters	<i>entier</i>	La valeur doit être supérieure à 0.
normalization_method	zscore minmax none	

Propriétés orakmeansnode	Valeurs	Description de la propriété
distance_function	Euclidean Cosine	
iterations	entier	0–20.*
conv_tolerance	nombre	0.0–0.5.*
split_criterion	Variance Size	La valeur par défaut est Variance.*
num_bins	entier	La valeur doit être supérieure à 0.*
block_growth	entier	1–5.*
min_pct_attr_support	nombre	0.0–1.0.*

* Propriété ignorée si mode est défini sur Simple.

NMF Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oranmfnode.

Propriétés oranmfnode	Valeurs	Description de la propriété
normalization_method	minmax none	
use_num_features	booléen	*
num_features	entier	0–1. La valeur par défaut est estimée à partir des données par l'algorithme.*
random_seed	nombre	*
num_iterations	entier	0–500.*
conv_tolerance	nombre	0.0–0.5.*
display_all_features	booléen	*

* Propriété ignorée si mode est défini sur Simple.

Apriori Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oraapriorinode.

Propriétés oraapriorinode	Valeurs	Description de la propriété
content_field	champ	
id_field	champ	
max_rule_length	entier	2–20.
min_confidence	nombre	0.0–1.0.
min_support	nombre	0.0–1.0.
use_transactional_data	booléen	

Description de longueur minimale d'Oracle (MDL)

Aucune propriété particulière n'est définie pour les noeuds du type oramdlnode. Reportez-vous aux propriétés Oracle courantes au début de cette section.

Importance de l'attribut d'Oracle (IA)

Les propriétés suivantes sont disponibles pour les nœuds du type `oraainode`.

Propriétés <code>oraainode</code>	Valeurs	Description de la propriété
<code>custom_fields</code>	<i>booléen</i>	Si la valeur est définie sur <code>true</code> (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le nœud actuel. Si elle est définie sur <code>false</code> (faux), les paramètres actuels provenant d'un nœud <code>Typet</code> en amont sont utilisés.
<code>selection_mode</code>	<code>ImportanceLevel</code> <code>Importance-Value</code> <code>TopN</code>	
<code>select_important</code>	<i>booléen</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceLevel</code> , indique si les champs importants doivent être sélectionnés.
<code>important_label</code>	<i>chaîne</i>	Indique l'étiquette du classement « significatif ».
<code>select_marginal</code>	<i>booléen</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceLevel</code> , indique si les champs marginaux doivent être sélectionnés.
<code>marginal_label</code>	<i>chaîne</i>	Indique l'étiquette du classement « marginal ».
<code>important_above</code>	<i>nombre</i>	0.0–1.0.
<code>select_unimportant</code>	<i>booléen</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceLevel</code> , indique si les champs non importants doivent être sélectionnés.
<code>unimportant_label</code>	<i>chaîne</i>	Indique l'étiquette du classement « non significatif ».
<code>unimportant_below</code>	<i>nombre</i>	0.0–1.0.
<code>importance_value</code>	<i>nombre</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>ImportanceValue</code> , indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 100.
<code>top_n</code>	<i>nombre</i>	Quand la propriété <code>selection_mode</code> est définie sur <code>TopN</code> , indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 1000.

Propriétés du nugget de modèle Oracle

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des modèles Oracle.

Oracle Naive Bayes

Aucune propriété particulière n'est définie pour les nœuds du type `applyoranbnode`.

Oracle Adaptive Bayes

Aucune propriété particulière n'est définie pour les nœuds du type `applyoraabnode`.

Oracle Support Vector Machines

Aucune propriété particulière n'est définie pour les noeuds du type applyorasvmnode.

Arbre décision Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type applyoradecisiontreenode.

Propriétés applyoradecisiontreenode	Valeurs	Description de la propriété
use_costs	booléen	
display_rule_ids	booléen	

O-Cluster Oracle

Aucune propriété particulière n'est définie pour les noeuds du type applyoraoclusternode.

KMeans Oracle

Aucune propriété particulière n'est définie pour les noeuds du type applyorakmeansnode.

NMF Oracle

La propriété suivante est disponible pour les noeuds du type applyoranmfnode :

Propriétés applyoranmfnode	Valeurs	Description de la propriété
display_all_features	booléen	

Apriori Oracle

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

MDL Oracle

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Propriétés de nœud pour la modélisation IBM DB2**Propriétés du nœud de modélisation IBM DB2**

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données IBM InfoSphere Warehouse (ISW).

Propriétés communes du nœud ISW	Valeurs	Description de la propriété
inputs	Liste des champs	
datasource		

Propriétés communes du nœud ISW	Valeurs	Description de la propriété
username		
password		
epassword		
enable_power_options	<i>booléen</i>	
power_options_max_memory	<i>entier</i>	La valeur doit être supérieure à 32.
power_options_cmdline	<i>chaîne</i>	
mining_data_custom_sql	<i>chaîne</i>	
logical_data_custom_sql	<i>chaîne</i>	
mining_settings_custom_sql		

Arbre décision ISW

Les propriétés suivantes sont disponibles pour les nœuds du type db2imtreenode.

Propriétés db2imtreenode	Valeurs	Description de la propriété
target	<i>champ</i>	
perform_test_run	<i>booléen</i>	
use_max_tree_depth	<i>booléen</i>	
max_tree_depth	<i>entier</i>	Valeur supérieure à 0.
use_maximum_purity	<i>booléen</i>	
maximum_purity	<i>nombre</i>	Nombre compris entre 0 et 100.
use_minimum_internal_cases	<i>booléen</i>	
minimum_internal_cases	<i>entier</i>	Valeur supérieure à 1.
use_costs	<i>booléen</i>	
costs	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>{{drugA drugB 1.5} {drugA drugC 2.1}}</code> , où les arguments entre <code>{}</code> sont les coûts prévus réels.

Association ISW

Les propriétés suivantes sont disponibles pour les nœuds du type db2imassocnode.

Propriétés db2imassocnode	Valeurs	Description de la propriété
use_transactional_data	<i>booléen</i>	
id_field	<i>champ</i>	
content_field	<i>champ</i>	
data_table_layout	<i>basic</i> <i>limited_length</i>	
max_rule_size	<i>entier</i>	La valeur doit être supérieure à 2.
min_rule_support	<i>nombre</i>	0–100%
min_rule_confidence	<i>nombre</i>	0–100%
use_item_constraints	<i>booléen</i>	
item_constraints_type	<i>Include</i> <i>Exclude</i>	
use_taxonomy	<i>booléen</i>	

Propriétés db2imassocnode	Valeurs	Description de la propriété
taxonomy_table_name	chaîne	Nom de la table DB2 dans laquelle stocker les détails de la taxonomie.
taxonomy_child_column_name	chaîne	Nom de la colonne enfant de la table de taxonomie. La colonne enfant comporte les noms des éléments ou des catégories.
taxonomy_parent_column_name	chaîne	Nom de la colonne parent de la table de taxonomie. La colonne parent contient les noms des catégories.
load_taxonomy_to_table	booléen	Permet de savoir si les informations sur la taxonomie stockées dans IBM® SPSS® Modeler doivent être envoyées à la table de taxonomie lors de la création du modèle. La table de taxonomie est supprimée si elle existe déjà. Les informations sur la taxonomie sont stockées avec le noeud de création du modèle et peuvent être modifiées via les boutons Modifier les catégories et Modifier la taxonomie.

Séquence ISW

Les propriétés suivantes sont disponibles pour les noeuds du type db2imsequencenode.

Propriétés db2imsequencenode	Valeurs	Description de la propriété
id_field	champ	
group_field	champ	
content_field	champ	
max_rule_size	entier	La valeur doit être supérieure à 2.
min_rule_support	nombre	0–100%
min_rule_confidence	nombre	0–100%
use_item_constraints	booléen	
item_constraints_type	Include Exclude	
use_taxonomy	booléen	
taxonomy_table_name	chaîne	Nom de la table DB2 dans laquelle stocker les détails de la taxonomie.
taxonomy_child_column_name	chaîne	Nom de la colonne enfant de la table de taxonomie. La colonne enfant comporte les noms des éléments ou des catégories.
taxonomy_parent_column_name	chaîne	Nom de la colonne parent de la table de taxonomie. La colonne parent contient les noms des catégories.
load_taxonomy_to_table	booléen	Permet de savoir si les informations sur la taxonomie stockées dans SPSS Modeler doivent être envoyées à la table de taxonomie lors de la création du modèle. La table de taxonomie est supprimée si elle existe déjà. Les informations sur la taxonomie sont stockées avec le noeud de création du modèle et peuvent être modifiées via les boutons Modifier les catégories et Modifier la taxonomie.

Régression ISW

Les propriétés suivantes sont disponibles pour les nœuds du type `db2imregnode`.

Propriétés <code>db2imregnode</code>	Valeurs	Description de la propriété
<code>target</code>	<i>champ</i>	
<code>regression_method</code>	transform linear polynomial rbf	Consultez le tableau suivant pour les propriétés s'appliquant uniquement si la <code>regression_method</code> est définie sur <code>rbf</code> .
<code>perform_test_run</code>	<i>champ</i>	
<code>limit_rsquared_value</code>	<i>booléen</i>	
<code>max_rsquared_value</code>	<i>nombre</i>	Valeur comprise entre 0,0 et 1,0.
<code>use_execution_time_limit</code>	<i>booléen</i>	
<code>execution_time_limit_mins</code>	<i>entier</i>	Valeur supérieure à 0.
<code>use_max_degree_polynomial</code>	<i>booléen</i>	
<code>max_degree_polynomial</code>	<i>entier</i>	
<code>use_intercept</code>	<i>booléen</i>	
<code>use_auto_feature_selection_method</code>	<i>booléen</i>	
<code>auto_feature_selection_method</code>	normal adjusted	
<code>use_min_significance_level</code>	<i>booléen</i>	
<code>min_significance_level</code>	<i>nombre</i>	
<code>use_min_significance_level</code>	<i>booléen</i>	

Les propriétés suivantes ne s'appliquent que si la `regression_method` est définie sur `rbf`.

Propriétés <code>db2imregnode</code>	Valeurs	Description de la propriété
<code>use_output_sample_size</code>	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
<code>output_sample_size</code>	<i>entier</i>	La valeur par défaut est de 2. Le minimum est 1.
<code>use_input_sample_size</code>	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
<code>input_sample_size</code>	<i>entier</i>	La valeur par défaut est de 2. Le minimum est 1.
<code>use_max_num_centers</code>	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
<code>max_num_centers</code>	<i>entier</i>	La valeur par défaut est de 20. Le minimum est 1.
<code>use_min_region_size</code>	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
<code>min_region_size</code>	<i>entier</i>	La valeur par défaut est de 15. Le minimum est 1.
<code>use_max_data_passes</code>	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
<code>max_data_passes</code>	<i>entier</i>	La valeur par défaut est de 5. Le minimum est 2.

use_min_data_passes	<i>booléen</i>	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
min_data_passes	<i>entier</i>	La valeur par défaut est de 5. Le minimum est 2.

Classification ISW

Les propriétés suivantes sont disponibles pour les noeuds du type `db2imclusternode`.

Propriétés <code>db2imclusternode</code>	Valeurs	Description de la propriété
cluster_method	demographic kohonen birch	
kohonen_num_rows	<i>entier</i>	
kohonen_num_columns	<i>entier</i>	
kohonen_passes	<i>entier</i>	
use_num_passes_limit	<i>booléen</i>	
use_num_clusters_limit	<i>booléen</i>	
max_num_clusters	<i>entier</i>	Valeur supérieure à 1.
birch_dist_measure	log_likelihood euclidean	La valeur par défaut est log_likelihood.
birch_num_cfleaves	<i>entier</i>	La valeur par défaut est de 1 000.
birch_num_refine_passes	<i>entier</i>	La valeur par défaut est 3, la valeur minimum est 1.
use_execution_time_limit	<i>booléen</i>	
execution_time_limit_mins	<i>entier</i>	Valeur supérieure à 0.
min_data_percentage	<i>nombre</i>	0–100%
use_similarity_threshold	<i>booléen</i>	
similarity_threshold	<i>nombre</i>	Valeur comprise entre 0,0 et 1,0.

Naive Bayes ISW

Les propriétés suivantes sont disponibles pour les noeuds du type `db2imnbsnode`.

Propriétés <code>db2imnbsnode</code>	Valeurs	Description de la propriété
perform_test_run	<i>booléen</i>	
probability_threshold	<i>nombre</i>	La valeur par défaut est 0.001. La valeur minimum est de 0 ; la valeur maximum est de 1,000.
use_costs	<i>booléen</i>	
costs	<i>structurées</i>	Propriété structurée utilisant la syntaxe : <code>{drugA drugB 1.5} {drugA drugC 2.1}</code> , où les arguments entre <code>{}</code> sont les coûts prévus réels.

Régression logistique ISW

Les propriétés suivantes sont disponibles pour les nœuds du type db2imlognode.

Propriétés db2imlognode	Valeurs	Description de la propriété
perform_test_run	booléen	
use_costs	booléen	
costs	structurées	Propriété structurée utilisant la syntaxe : <code>[[drugA drugB 1.5] {drugA drugC 2.1}]</code> , où les arguments entre <code>{ }</code> sont les coûts prévus réels.

Séries temporelles ISW

Remarque : Le paramètre des champs d'entrée n'est pas utilisé pour ce nœud. Si le paramètre des champs d'entrée se trouve dans le script, un avertissement apparaît et signale que le nœud contient les champs entrants *heure* et *cibles* mais pas de champs d'entrée.

Les propriétés suivantes sont disponibles pour les nœuds du type db2imtimeseriesnode.

Propriétés db2imtime-seriesnode	Valeurs	Description de la propriété
time	champ	Entier, heure ou date autorisé.
targets	liste des champs	
forecasting_algorithm	arima exponen- tial_smoothing sea- sonal_trend_de- composition	
forecasting_end_time	auto integer date time	
use_records_all	booléen	Si faux, use_records_start et use_records_end doivent être définis.
use_records_start	entier / heure / date	Dépend du type du champ Temps
use_records_end	entier / heure / date	Dépend du type du champ Temps
interpolation_method	none linear exponen- tial_splines cubic_splines	

Propriétés du nugget de modèle IBM DB2

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des modèles IBM DB2 ISW.

Arbre décision ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imtree`node.

Association ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Séquence ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Régression ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imreg`node.

Classification ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imcluster`node.

Naive Bayes ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imnb`node.

Régression logistique ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imlog`node.

Séries temporelles ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Propriétés du noeud pour la modélisation IBM Netezza Analytics**Propriétés des noeuds de modélisation Netezza**

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données IBM Netezza.

Propriétés communes des noeuds Netezza	Valeurs	Description de la propriété
<code>custom_fields</code>	<i>booléen</i>	Si la valeur est définie sur <code>true</code> (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur <code>false</code> (faux), les paramètres actuels provenant d'un noeud <code>Type</code> en amont sont utilisés.
<code>inputs</code>	<i>[champ1 ... champN]</i>	Champs d'entrée ou variable indépendante utilisés par le modèle.

Propriétés communes des nœuds Netezza	Valeurs	Description de la propriété
target	<i>champ</i>	Champ cible (continu ou qualitatif).
record_id	<i>champ</i>	Champ à utiliser comme identifiant d'enregistrement unique.
use_upstream_connection	<i>booléen</i>	Si vrai (valeur par défaut), les détails de connexion spécifiés dans un nœud en amont. Non utilisé si <code>move_data_to_connection</code> est spécifié.
move_data_connection	<i>booléen</i>	Si vrai, déplace les données vers la base de données mentionnée par <code>connection</code> . Non utilisé si <code>use_upstream_connection</code> est spécifié.
connection	<i>structurées</i>	La chaîne de connexion à la base de données Netezza où le modèle est stocké. Propriété structurée utilisant la syntaxe : <code>['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attrbs>' {true false}]</code> où : <dsn> est le nom de la source de données. <username> et <psw> sont le nom d'utilisateur et le mot de passe de la base de données <catname> est le nom du catalogue <conn_attrbs> sont les attributs de connexion true false indique si un mot de passe est requis.
table_name	<i>chaîne</i>	Le nom de la table de la base de données où le modèle est stocké.
use_model_name	<i>booléen</i>	Si vrai, utilise le nom spécifié par <code>model_name</code> comme nom du modèle, sinon le nom du modèle est créé par le système.
model_name	<i>chaîne</i>	Nom personnalisé du nouveau modèle.
include_input_fields	<i>booléen</i>	Si vrai, passe tous les champs d'entrée en aval, sinon passe uniquement le <code>record_id</code> et les champs générés par le modèle.

Arbre décision Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type `netezzadectreenode`.

Propriétés <code>netezzadectreenode</code>	Valeurs	Description de la propriété
impurity_measure	Entropy Gini	La mesure d'impureté, utilisée pour évaluer le meilleur endroit où diviser l'arbre.
max_tree_depth	<i>entier</i>	Nombre de niveaux maximum que l'arbre peut atteindre. La valeur par défaut est 62 (le nombre maximum possible).
min_improvement_splits	<i>nombre</i>	Amélioration minimum de l'impureté pour que la division ait lieu. La valeur par défaut est de 0,01.
min_instances_split	<i>entier</i>	Nombre minimum d'enregistrements non divisés restants avant que la division n'ait lieu. La valeur par défaut est 2 (le nombre minimum possible).

Propriétés netezzadectreenode	Valeurs	Description de la propriété
weights	<i>structurées</i>	Pondérations relatives des classes. Propriété structurée utilisant la syntaxe : set :netezza_dectree.weights = [{drugA 0.3}{drugB 0.6}] La valeur par défaut est une pondération de 1 pour toutes les classes.
pruning_measure	Acc wAcc	La valeur par défaut est Acc (précision). La wAcc (précision pondérée) tient compte des pondérations de classe tout en appliquant l'élagage.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	La valeur par défaut est d'utiliser allTrainingData pour estimer la précision du modèle. Utilisez partitionTrainingData pour spécifier un pourcentage de données d'apprentissage à utiliser ou useOtherTable pour utiliser un ensemble de données d'apprentissage d'une table de base de données spécifique.
perc_training_data	<i>nombre</i>	Si les prune_tree_options sont définies sur partitionTrainingData, spécifie le pourcentage de données à utiliser pour l'apprentissage.
prune_seed	<i>entier</i>	Graine aléatoire à utiliser pour dupliquer les résultats d'analyse quand les prune_tree_options sont définies sur partitionTrainingData ; la valeur par défaut est 1.
pruning_table	<i>chaîne</i>	Nom de la table d'un ensemble de données d'élagage distinct pour estimer la précision du modèle.
compute_probabilities	<i>booléen</i>	Si vrai, produit un champ de niveau de confiance (probabilité) ainsi que le champ de prévision.

Netezza K-means

Les propriétés suivantes sont disponibles pour les noeuds du type netezzakmeansnode.

Propriétés netezzakmeansnode	Valeurs	Description de la propriété
distance_measure	Euclidean Manhattan Canberra maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
num_clusters	<i>entier</i>	Le nombre de classes à créer ; la valeur par défaut est 3.
max_iterations	<i>entier</i>	Le nombre d'itérations d'algorithme après lequel arrêter l'apprentissage du modèle ; la valeur par défaut est 5.
rand_seed	<i>entier</i>	Graine aléatoire à utiliser pour dupliquer les résultats d'analyse ; la valeur par défaut est 12345.

Bayes Net Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type `netezabayesnode`.

Propriétés <code>netezabayesnode</code>	Valeurs	Description de la propriété
<code>base_index</code>	<i>entier</i>	Identifiant numérique attribué au premier champ d'entrée pour la gestion interne ; la valeur par défaut est 777.
<code>sample_size</code>	<i>entier</i>	Taille de l'échantillon à prélever si le nombre d'attributs est très grand ; la valeur par défaut est 10 000.
<code>display_additional_information</code>	<i>booléen</i>	Si vrai, affiche les informations de progression supplémentaires dans une boîte de dialogue.
<code>type_of_prediction</code>	<code>best neighbors</code> <code>nn-neighbors</code>	Type d'algorithme de prévision à utiliser : meilleur (voisin le plus corrélé), voisins (prévision pondérée des voisins) ou voisins nn (voisins non nuls).

Naive Bayes Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type `netezanaivebayesnode`.

Propriétés <code>netezanaive-bayesnode</code>	Valeurs	Description de la propriété
<code>compute_probabilities</code>	<i>booléen</i>	Si vrai, produit un champ de niveau de confiance (probabilité) ainsi que le champ de prévision.
<code>use_m_estimation</code>	<i>booléen</i>	Si vrai, utilise la technique de m-estimation pour éviter les probabilités zéro pendant l'estimation.

KNN Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type `netezaknnnode`.

Propriétés <code>netezaknnnode</code>	Valeurs	Description de la propriété
<code>weights</code>	<i>structurées</i>	Propriété structurée utilisée pour attribuer des pondérations aux classes individuelles. Exemple : <code>set :netezaknnnode.weights = {{drugA 0.3}{drugB 0.6}}</code>
<code>distance_measure</code>	Euclidean Manhattan Canberra Maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
<code>num_nearest_neighbors</code>	<i>entier</i>	Nombre de voisins les plus proches pour un cas particulier ; la valeur par défaut est 3.
<code>standardize_measurements</code>	<i>booléen</i>	Si vrai, standardise les mesures des champs d'entrée continus avant de calculer les valeurs de distance.
<code>use_coresets</code>	<i>booléen</i>	Si vrai, utilise l'échantillonnage d'ensembles principaux pour accélérer le calcul de grands ensembles de données.

Classification par division Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type `netezadivclusternode`.

Propriétés <code>netezadivclusternode</code>	Valeurs	Description de la propriété
<code>distance_measure</code>	Euclidean Manhattan Canberra Maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
<code>max_iterations</code>	<i>entier</i>	Nombre maximum d'itérations d'algorithme à effectuer avant l'arrêt de l'apprentissage du modèle ; la valeur par défaut est 5.
<code>max_tree_depth</code>	<i>entier</i>	Nombre maximum de niveaux dans lesquels les ensembles de données peuvent être divisés ; la valeur par défaut est 3.
<code>rand_seed</code>	<i>entier</i>	Graine aléatoire utilisée pour dupliquer les analyses ; la valeur par défaut est 12345.
<code>min_instances_split</code>	<i>entier</i>	Nombre minimum d'enregistrements pouvant être divisés, la valeur par défaut est 5.
<code>level</code>	<i>entier</i>	Niveau hiérarchique selon lequel les enregistrements sont évalués ; la valeur par défaut est -1.

ACP Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type `netezapcanode`.

Propriétés <code>netezapcanode</code>	Valeurs	Description de la propriété
<code>center_data</code>	<i>booléen</i>	Si vrai (par défaut), effectue un centrage des données (également nommé « soustraction moyenne ») avant l'analyse.
<code>perform_data_scaling</code>	<i>booléen</i>	Si vrai, effectue la mise à l'échelle des données avant l'analyse. Ceci peut rendre l'analyse moins arbitraire quand différentes variables sont mesurées dans différentes unités.
<code>force_eigensolve</code>	<i>booléen</i>	Si vrai, utilise une méthode moins précise mais plus rapide pour trouver les composantes principales.
<code>pc_number</code>	<i>entier</i>	Nombre de composantes principales auquel l'ensemble de données doit être réduit ; la valeur par défaut est 1.

Arbre de régression Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type `netezaregtreenode`.

Propriétés <code>netezaregtreenode</code>	Valeurs	Description de la propriété
<code>max_tree_depth</code>	<i>entier</i>	Nombre maximum de niveaux auxquels l'arbre peut grandir sous le noeud racine ; la valeur par défaut est 10.
<code>split_evaluation_measure</code>	Variance	Mesure d'impureté de la classe, utilisée pour évaluer le meilleur endroit pour diviser l'arbre ; la valeur par défaut (et actuellement la seule option) est Variance.

Propriétés netezzaregreesnode	Valeurs	Description de la propriété
min_improvement_splits	<i>nombre</i>	Quantité minimale pour réduire l'impureté avant de créer une nouvelle division dans l'arbre.
min_instances_split	<i>entier</i>	Le nombre minimal d'enregistrements pouvant être divisés.
pruning_measure	mse r2 pearson spearman	Méthode à utiliser pour l'élagage.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	La valeur par défaut est d'utiliser allTrainingData pour estimer la précision du modèle. Utilisez partitionTrainingData pour spécifier un pourcentage de données d'apprentissage à utiliser ou useOtherTable pour utiliser un ensemble de données d'apprentissage d'une table de base de données spécifique.
perc_training_data	<i>nombre</i>	Si les prune_tree_options sont définies sur PercTrainingData, spécifie le pourcentage de données à utiliser pour l'apprentissage.
prune_seed	<i>entier</i>	Graine aléatoire à utiliser pour dupliquer les résultats d'analyse quand les prune_tree_options sont définies sur PercTrainingData ; la valeur par défaut est 1.
pruning_table	<i>chaîne</i>	Nom de la table d'un ensemble de données d'élagage distinct pour estimer la précision du modèle.
compute_probabilities	<i>booléen</i>	Si vrai, spécifie que les variances des classes attribuées doivent être incluses dans la sortie.

Régression linéaire Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type netezzalineressionnode.

Propriétés netezzalineressionnode	Valeurs	Description de la propriété
use_svd	<i>booléen</i>	Si vrai, utilise la matrice de décomposition en valeurs singulières à la place de la matrice d'origine, pour améliorer la vitesse et la précision numérique.
include_intercept	<i>booléen</i>	Si vrai (défaut), augmente la précision générale de la solution.
calculate_model_diagnostics	<i>booléen</i>	Si vrai, calcule le diagnostic sur le modèle.

Séries temporelles Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type netezzatimeseriesnode.

Propriétés netezzatimeseriesnode	Valeurs	Description de la propriété
time_points	<i>champ</i>	Champ d'entrée contenant les valeurs de date et d'heure des séries temporelles.
time_series_ids	<i>champ</i>	Un champ d'entrée contenant des ID de séries temporelles à utiliser si l'entrée contient plusieurs séries temporelles.

Propriétés netezza-time-seriesnode	Valeurs	Description de la propriété
model_table	<i>champ</i>	Nom de la table de la base de données dans laquelle le modèle de séries temporelles Netezza sera stocké.
description_table	<i>champ</i>	Nom de la table d'entrée qui contient les noms et descriptions des séries temporelles.
seasonal_adjustment_table	<i>champ</i>	Nom de la table de sortie dans laquelle les valeurs ajustées de manière saisonnière calculées par des algorithmes de lissage exponentiel ou de décomposition des tendances saisonnières seront stockées.
algorithm_name	SpectralAnalysis ou spectral ExponentialSmoothing ou esmoothing ARIMA SeasonalTrend-Decomposition ou std	Algorithme à utiliser pour la modélisation des séries temporelles.
trend_name	<i>chaîne</i>	Type de tendance du lissage exponentiel.
seasonality_type	<i>chaîne</i>	Type de saisonnalité pour le lissage exponentiel.
interpolation_method	linear cubic spline exponential-spline	Méthode d'interpolation à utiliser.
earliest_time	<i>champ</i>	Heure de commencement, si une partie des séries temporelles est utilisée.
latest_time	<i>champ</i>	Heure de fin, si une partie des séries temporelles est utilisée.
p	<i>entier</i>	ARIMA - degrés d'autocorrélation non saisonniers.
q	<i>entier</i>	ARIMA - valeur de calcul non saisonnier.
d	<i>entier</i>	ARIMA - nombre d'ordres de moyenne mobile non saisonniers dans le modèle.
sp	<i>entier</i>	ARIMA - degrés d'autocorrélation saisonniers.
sq	<i>entier</i>	ARIMA - valeur de dérivation saisonnière.
sd	<i>entier</i>	ARIMA - nombre d'ordres de moyenne mobile saisonniers dans le modèle.
period	<i>entier</i>	Longueur du cycle saisonnier spécifié en association avec <code>units_period</code> . Ne s'applique pas à l'analyse spectrale.
units_period	Milliseconds Seconds Minutes Hours Days Weeks Quarters Years	Unités dans lesquelles <code>period</code> est exprimé. Par exemple, pour une série temporelle, utilisez 1 pour <code>period</code> et <code>Weeks</code> pour <code>units_period</code> .
include_history	<i>booléen</i>	Indique si les valeurs historiques doivent être incluses dans la sortie.
include_interpolated_values	<i>booléen</i>	Indique si les valeurs interpolées doivent être incluses dans la sortie. Ne s'applique pas si <code>include_history</code> est <i>false</i> .

Linéaire généralisé Netezza

Les propriétés suivantes sont disponibles pour les nœuds du type `netezzaglmnode`.

Propriétés <code>netezzaglmnode</code>	Valeurs	Description de la propriété
<code>response_variable_distribution</code>	bernoulli gaussian poisson binomial negativebino- mial wald gamma	Type de distribution ; le type par défaut est bernoulli.
<code>distribution_parameter</code>	<i>nombre</i>	Valeur de paramètre de distribution à utiliser. S'applique uniquement si <code>distribution</code> est <code>Negativebinomial</code> .
<code>trials</code>	<i>entier</i>	S'applique uniquement si <code>distribution</code> est <code>Binomial</code> . Lorsque la réponse cible est un nombre d'événements se produisant dans un ensemble d'essais, le champ <code>target</code> contient le nombre d'événements et le champ <code>trials</code> contient le nombre d'essais.
<code>model_table</code>	<i>champ</i>	Nom de la table de la base de données dans laquelle le modèle linéaire généralisé Netezza sera stocké.
<code>max_iterations</code>	<i>entier</i>	Nombre maximum d'itérations que l'algorithme doit exécuter ; la valeur par défaut est de 20.
<code>max_error</code>	<i>nombre</i>	Valeur du nombre maximal d'erreurs (en notation scientifique) à laquelle l'algorithme doit s'arrêter de rechercher le modèle le plus adapté. La valeur par défaut est de -3, soit 1E-3 ou 0,001.
<code>error_threshold</code>	<i>nombre</i>	Valeur (en notation scientifique) sous laquelle les erreurs sont traitées comme ayant une valeur de zéro. La valeur par défaut est de -7, ce qui signifie que les valeurs d'erreurs inférieures à 1E-7 (ou 0,0000001) sont considérées comme non significatives.
<code>link_function</code>	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	Fonction de lien à utiliser ; la fonction par défaut est <code>logit</code> .

Propriétés netezzaglmnode	Valeurs	Description de la propriété
link_function_parameter	<i>nombre</i>	Valeur de paramètre de fonction de lien à utiliser. S'applique uniquement si link_function est power ou oddspower.
intercept	<i>booléen</i>	Si true, inclut la constante dans le modèle.

Propriétés du nugget de modèle Netezza

Les propriétés suivantes sont communes aux noeuds de modèle de base de données Netezza.

Propriétés communes du nugget de modèle Netezza	Valeurs	Description de la propriété
connection	<i>chaîne</i>	La chaîne de connexion à la base de données Netezza où le modèle est stocké.
table_name	<i>chaîne</i>	Le nom de la table de la base de données où le modèle est stocké.

Les autres propriétés du nugget de modèle sont les mêmes que celles du noeud de modélisation correspondant.

Les noms de script des nuggets de modèle sont les suivants.

Nugget de modèle	Nom de script
Arbre de décision	applynetezadectreenode
K-means	applynetezzakmeansnode
Bayes Net	applynetezabayesnode
Naive Bayes	applynetezanaivebayesnode
KNN	applynetezaknnnode
Classification par division	applynetezadivclusternode
ACP	applynetezapcanode
Arbre de régression	applynetezaregtreenode
Régression linéaire	applynetezalineregressionnode
Séries temporelles	applynetezatimeseriesnode
Linéaire généralisé	applynetezaglmnode

Propriétés des noeuds de sortie

Les propriétés de nœud de sortie diffèrent légèrement de celles des autres types de nœud. Au lieu de faire référence à une option de nœud particulière, les propriétés du nœud de sortie stockent une référence à l'objet de sortie. Cela peut être utile lorsque vous choisissez une valeur dans un tableau pour la définir en tant que paramètre de flux.

Cette section décrit les propriétés de génération de scripts disponibles pour les noeuds de sortie.

Propriétés de analysisnode



Le nœud Analyse évalue la capacité des modèles prédictifs à générer des prévisions précises. Les nœuds Analyse comparent les valeurs prédites et les valeurs réelles d'un ou de plusieurs nuggets de modèle. Ils peuvent également comparer entre eux les modèles prédictifs. [Pour plus d'informations, reportez-vous à la section Nœud Analyse dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modèles 15.](#)

Exemple

```
create analysisnode
#"Analysis" tab
set :analysisnode.coincidence = True
set :analysisnode.performance = True
set :analysisnode.confidence = True
set :analysisnode.threshold = 75
set :analysisnode.improve_accuracy = 3
set :analysisnode.inc_user_measure = True
#"Define User Measure..."
set :analysisnode.user_if = "@TARGET = @PREDICTED"
set :analysisnode.user_then = "101"
set :analysisnode.user_else = "1"
set :analysisnode.user_compute = [Mean Sum]
set :analysisnode.by_fields = ['Drug']
#"Output" tab
set :analysisnode.output_format = HTML
set :analysisnode.full_filename = "C:/output/analysis_out.html"
```

propriétés analysisnode	Le type de données	Description de la propriété
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du nœud de sortie.
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name est vrai, indique le nom à utiliser.

propriétés analysisnode	Le type de données	Description de la propriété
output_format	Text (.txt) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
by_fields	[champ champ champ]	
full_filename	chaîne	Si vous indiquez Disque, Données ou HTML : nom du fichier de sortie utilisé.
coincidence	Booléen	
performance	Booléen	
confidence	Booléen	
threshold	nombre	
improve_accuracy	nombre	
inc_user_measure	Booléen	
user_if	expr	
user_then	expr	
user_else	expr	
user_compute	[Mean Sum Min Max SDev]	

Propriétés de dataauditnode



Le noeud Audit données fournit un premier aperçu complet des données, notamment des statistiques récapitulatives, des histogrammes et distributions pour chaque champ, ainsi que des informations sur les valeurs éloignées, les valeurs manquantes et les valeurs extrêmes. Les résultats sont affichés dans une matrice facile à lire pouvant être triée et utilisée pour générer les noeuds de préparation des données et des graphiques grandeur nature. [Pour plus d'informations, reportez-vous à la section Noeud Audit données dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create dataauditnode
connect :variablefilenode to :dataauditnode
set :dataauditnode.custom_fields = True
set :dataauditnode.fields = [Age Na K]
set :dataauditnode.display_graphs = True
set :dataauditnode.basic_stats = True
set :dataauditnode.advanced_stats = True
set :dataauditnode.median_stats = False
set :dataauditnode.calculate = [Count Breakdown]
set :dataauditnode.outlier_detection_method = std
set :dataauditnode.outlier_detection_std_outlier = 1.0
set :dataauditnode.outlier_detection_std_extreme = 3.0
```

```
set :dataauditnode.output_mode = Screen
```

propriétés dataauditnode	Le type de données	Description de la propriété
custom_fields	Booléen	
fields	[champ1 ... champN]	
overlay	champ	
display_graphs	Booléen	Permet d'activer ou de désactiver l'affichage des graphiques dans la matrice de sortie.
basic_stats	Booléen	
advanced_stats	Booléen	
median_stats	Booléen	
calculate	Count Breakdown	Utilisée pour calculer les valeurs manquantes. Sélectionnez une méthode de calcul, les deux ou bien aucune.
outlier_detection_method	std iqr	Utilisé pour indiquer la méthode de détection des valeurs éloignées et des valeurs extrêmes.
outlier_detection_std_outlier	nombre	Si outlier_detection_method est défini sur std, indique le nombre à utiliser pour définir les valeurs éloignées.
outlier_detection_std_extreme	nombre	Si outlier_detection_method est défini sur std, indique le nombre à utiliser pour définir les valeurs extrêmes.
outlier_detection_iqr_outlier	nombre	Si outlier_detection_method est défini sur iqr, indique le nombre à utiliser pour définir les valeurs éloignées.
outlier_detection_iqr_extreme	nombre	Si outlier_detection_method est défini sur iqr, indique le nombre à utiliser pour définir les valeurs extrêmes.
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name est vrai, indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
paginate_output	Booléen	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.

propriétés dataauditnode	Le type de données	Description de la propriété
lines_per_page	nombre	Lorsque cette propriété est utilisée avec <code>paginate_output</code> , elle indique le nombre de lignes par page de sortie.
full_filename	chaîne	

Propriétés de matrixnode



Le noeud Matrice permet de créer un tableau dans lequel les relations entre les champs sont indiquées. Il s'agit généralement de deux champs symboliques, mais il peut également s'agir de champs booléens ou numériques. [Pour plus d'informations, reportez-vous à la section Noeud Matrice dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create matrixnode
# Onglet Paramètres
set :matrixnode.fields = Numerics
set :matrixnode.row = 'K'
set :matrixnode.column = 'Na'
set :matrixnode.cell_contents = Fonction
set :matrixnode.function_field = 'Age'
set :matrixnode.function = Sum
# Onglet Apparence
set :matrixnode.sort_mode = Ascending
set :matrixnode.highlight_top = 1
set :matrixnode.highlight_bottom = 5
set :matrixnode.display = [Counts Expected Residuals]
set :matrixnode.include_totals = True
# Onglet Sortie
set :matrixnode.full_filename = "C:/output/matrix_output.html"
set :matrixnode.output_format = HTML
set :matrixnode.paginate_output = true
set :matrixnode.lines_per_page = 50
```

propriétés matrixnode	Le type de données	Description de la propriété
fields	Selected Flags Numerics	
row	champ	
column	champ	
include_missing_values	Booléen	Indique si les valeurs manquantes de l'utilisateur (blancs) et du système (valeurs nulles) sont incluses dans la sortie des lignes et des colonnes.
cell_contents	CrossTabs Fonction	
function_field	chaîne	

propriétés matrixnode	Le type de données	Description de la propriété
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	<i>nombre</i>	Si non égal à zéro, alors True (vrai).
highlight_bottom	<i>nombre</i>	Si non égal à zéro, alors True (vrai).
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	<i>Booléen</i>	
use_output_name	<i>Booléen</i>	Indique si un nom de sortie personnalisé est utilisé.
output_name	<i>chaîne</i>	Si use_output_name est vrai, indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie. Les formats Formatted et Delimited peuvent utiliser le modificateur transposed, qui transpose les lignes et les colonnes dans le tableau. Par exemple : NODE.output_format=transposed Delimited
paginate_output	<i>Booléen</i>	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	<i>nombre</i>	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
full_filename	<i>chaîne</i>	

Propriétés de meansnode



Le noeud Moyennes compare les moyennes de groupes indépendants ou de paires de champs associés, afin de détecter toute différence sensible. Par exemple, vous pouvez comparer les revenus moyens avant et après l'application d'une augmentation, ou comparer les revenus des personnes ayant obtenu une augmentation avec ceux des personnes qui n'en ont pas eu. [Pour plus d'informations, reportez-vous à la section Noeud Moyennes dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create meansnode
set :meansnode.means_mode = BetweenFields
set :meansnode.paired_fields = [{'OPEN_BAL' 'CURR_BAL'}]
set :meansnode.label_correlations = true
set :meansnode.output_view = Advanced
set :meansnode.output_mode = File
set :meansnode.output_format = HTML
set :meansnode.full_filename = "C:/output/means_output.html"
```

propriétés meansnode	Le type de données	Description de la propriété
means_mode	BetweenGroups BetweenFields	Indique le type de statistique de moyenne à exécuter sur les données.
test_fields	[field1 ... fieldn]	Indique le champ de test lorsque means_mode est paramétré sur BetweenGroups.
grouping_field	champ	Indique le champ de regroupement.
paired_fields	{{field1 field2} {field3 field4} ...}	Indique les paires de champs à utiliser lorsque means_mode est paramétré sur BetweenFields.
label_correlations	Booléen	Indique si les étiquettes de corrélation apparaissent dans la sortie. Ce paramètre est applicable uniquement quand l'option means_mode est définie sur BetweenFields.
correlation_mode	Probability Absolute	Indique si les corrélations doivent être étiquetées par probabilité ou valeur absolue.
weak_label	chaîne	
medium_label	chaîne	
strong_label	chaîne	
weak_below_probability	nombre	Quand la propriété correlation_mode est définie sur Probability, indique la valeur de césure à utiliser pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).

propriétés meansnode	Le type de données	Description de la propriété
strong_above_probability	nombre	Valeur de césure utilisée pour les corrélations fortes.
weak_below_absolute	nombre	Quand la propriété <code>correlation_mode</code> est définie sur <code>Absolute</code> , indique la valeur de césure à utiliser pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_absolute	nombre	Valeur de césure utilisée pour les corrélations fortes.
unimportant_label	chaîne	
marginal_label	chaîne	
important_label	chaîne	
unimportant_below	nombre	Valeur de césure utilisée pour une importance de champ faible. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
important_above	nombre	
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Nom à utiliser.
output_mode	Screen File	Indique l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Indique le type de sortie.
full_filename	chaîne	
output_view	Simple Advanced	Indique si la vue simple ou avancée est affichée dans la sortie.

Propriétés de reportnode



Ce noeud permet de créer des rapports formatés contenant du texte fixe et des données, ainsi que des expressions calculées à partir de ces dernières. Le format du rapport est déterminé par des modèles texte définissant la structure du texte fixe et de la sortie de données. Vous pouvez définir un formatage de texte personnalisé en utilisant des balises HTML dans le modèle et en définissant des options dans l'onglet Sortie. Vous pouvez inclure des valeurs de données et d'autres sorties conditionnelles à l'aide des expressions CLEM du modèle. [Pour plus d'informations, reportez-vous à la section Noeud Rapport dans le chapitre 6 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create reportnode
set :reportnode.output_format = HTML
```

```

set :reportnode.full_filename = "C:/report_output.html"
set :reportnode.lines_per_page = 50
set :reportnode.title = "Report node created by a script"
set :reportnode.highlights = False

```

propriétés reportnode	Le type de données	Description de la propriété
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	HTML (.html) Text (.txt) Output (.cou)	Permet d'indiquer le type de sortie.
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name est vrai, indique le nom à utiliser.
text	chaîne	
full_filename	chaîne	
highlights	Booléen	
title	chaîne	
lines_per_page	nombre	

Propriétés de setglobalsnode



Le nœud V. globales (Valeurs globales) analyse les données et calcule des valeurs récapitulatives pouvant être utilisées dans des expressions CLEM. Par exemple, vous pouvez utiliser ce noeud pour calculer les statistiques d'un champ *âge*, puis utiliser la moyenne globale du champ *age* dans des expressions CLEM en insérant la fonction @GLOBAL_MEAN(*age*). [Pour plus d'informations, reportez-vous à la section Nœud V. globales \(Valeurs globales\) dans le chapitre 6 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```

create setglobalsnode
connect :typenode to :setglobalsnode
set :setglobalsnode.globals.Na = [Max Sum Mean]
set :setglobalsnode.globals.K = [Max Sum Mean]
set :setglobalsnode.globals.Age = [Max Sum Mean SDev]
set :setglobalsnode.clear_first = False
set :setglobalsnode.show_preview = True

```

propriétés setglobalsnode	Le type de données	Description de la propriété
globals	[Sum Mean Min Max SDev]	Propriété structurée où les champs à définir doivent être référencés avec la syntaxe suivante : set :setglobalsnode.globals.Age = [Sum Mean Min Max SDev]
clear_first	Booléen	
show_preview	Booléen	

Propriétés de statisticsnode



Le noeud Statistiques fournit des informations récapitulatives de base sur les champs numériques. Il calcule les statistiques récapitulatives des champs individuels et des corrélations entre les champs. [Pour plus d'informations, reportez-vous à la section Noeud Statistiques dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create statisticsnode
# "Settings" tab
set :statisticsnode.examine = ['Age' 'BP' 'Drug']
set :statisticsnode.statistics = [Mean Sum SDev]
set :statisticsnode.correlate = ['BP' 'Drug']
# "Correlation Labels..." section
set :statisticsnode.label_correlations = True
set :statisticsnode.weak_below_absolute = 0.25
set :statisticsnode.weak_label = "lower quartile"
set :statisticsnode.strong_above_absolute = 0.75
set :statisticsnode.medium_label = "middle quartiles"
set :statisticsnode.strong_label = "upper quartile"
# "Output" tab
set :statisticsnode.full_filename = "c:/output/statistics_output.html"
set :statisticsnode.output_format = HTML
```

propriétés statisticsnode	Le type de données	Description de la propriété
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name est vrai, indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Text (.txt) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
full_filename	chaîne	
examine	[champ champ champ]	
correlate	[champ champ champ]	
statistics	[Count Mean Sum Min Max Range Variance SDev SErr Median Mode]	
correlation_mode	Probability Absolute	Indique si les corrélations doivent être étiquetées par probabilité ou valeur absolue.
label_correlations	Booléen	
weak_label	chaîne	

propriétés statisticsnode	Le type de données	Description de la propriété
medium_label	chaîne	
strong_label	chaîne	
weak_below_probability	nombre	Quand la propriété <code>correlation_mode</code> est définie sur <code>Probability</code> , indique la valeur de césure à utiliser pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_probability	nombre	Valeur de césure utilisée pour les corrélations fortes.
weak_below_absolute	nombre	Quand la propriété <code>correlation_mode</code> est définie sur <code>Absolute</code> , indique la valeur de césure à utiliser pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_absolute	nombre	Valeur de césure utilisée pour les corrélations fortes.

Propriétés de statisticsoutputnode



Le noeud Sortie Statistics vous permet d'appeler une procédure IBM® SPSS® Statistics pour analyser les données IBM® SPSS® Modeler. De nombreuses procédures d'analyses SPSS Statistics sont disponibles. Ce noeud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Noeud Sortie Statistics dans le chapitre 8 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Les propriétés de ce noeud sont décrites dans [Propriétés de statisticsoutputnode](#) sur p. 318.

Propriétés de tablenode



Le noeud Table affiche les données au format tabulaire (ces données peuvent également être écrites dans un fichier). Ainsi, vous pouvez passer en revue les valeurs de données ou les exporter dans un format facilement lisible. [Pour plus d'informations, reportez-vous à la section Noeud Table dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create tablenode
set :tablenode.highlight_expr = "Age > 30"
set :tablenode.output_format = HTML
set :tablenode.transpose_data = true
set :tablenode.full_filename = "C:/output/table_output.htm"
set :tablenode.paginate_output = true
```

set :tablnode.lines_per_page = 50

propriétés tablnode	Le type de données	Description de la propriété
full_filename	chaîne	Si vous indiquez Disque, Données ou HTML : nom du fichier de sortie utilisé.
use_output_name	Booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name est vrai, indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
transpose_data	Booléen	Transpose les données avant l'exportation, afin que les lignes et les colonnes représentent respectivement les champs et les enregistrements.
paginate_output	Booléen	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	nombre	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
highlight_expr	chaîne	
output	chaîne	Propriété en lecture seule qui comporte une référence à la dernière table créée par le noeud.
value_labels	[{ Chaîneétiquette Valeur} { Chaîneétiquette Valeur} ...]	Permet d'attribuer des étiquettes aux paires de valeurs. Par exemple, set :typenode.value_labels. 'Drug'={{drugA label1} {drugB label2}}
display_places	entier	Définit le nombre de décimales du champ lorsqu'il est affiché (s'applique uniquement aux champs dont le stockage est Réel). La valeur -1 utilise le flux par défaut. Syntaxe : NODE.display_places. FIELDNAME
export_places	entier	Définit le nombre de décimales du champ lorsqu'il est exporté (s'applique uniquement aux champs dont le stockage est Réel). La valeur -1 utilise le flux par défaut. Syntaxe : NODE.export_places.FIELDNAME

propriétés tablenode	Le type de données	Description de la propriété
decimal_separator	DEFAULT PERIOD COMMA	Définit le séparateur décimal du champ (s'applique uniquement aux champs dont le stockage est Réel). Syntaxe : NODE.decimal_separator. FIELDNAME
date_format	"JJMAAA" "MMJJAA" "AAMMJJ" "AAAAMMJJ" AAAAJJJ DAY MONTH "JJ-MM-AA" "JJ-MM-AAAA" "MM-JJ-AA" "MM-JJ-AAAA" "JJ-MOI-AA" "JJ-MOI-AAAA" "AAAA-MM-JJ" "JJ.MM.AA" "JJ.MM.AAAA" "MM.JJ.AA" "MM.JJ.AAAA" "JJ.MOI.AA" "JJ.MOI.AAAA" "JJ/MM/AA" "JJ/MM/AAAA" "MM/JJ/AA" "MM/JJ/AAAA" "JJ/MOI/AA" "JJ/MOI/AAAA" MOIS AAAA t T AAAA ss SM AAAA	Définit le format de date du champ (s'applique uniquement aux champs dont le stockage est DATE ou TIMESTAMP). Syntaxe : NODE.date_format.FIELDNAME Par exemple, set :tablenode.date_format. 'LaunchDate' = "DDMMYY"
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	Définit le format horaire du champ (s'applique uniquement aux champs dont le stockage est TIME ou TIMESTAMP). Syntaxe : NODE.time_format.FIELDNAME Par exemple, set :tablenode.time_format. set 'BOF_enter' = "HHMMSS"

propriétés tablenode	Le type de données	Description de la propriété
column_width	entier	Définit la largeur des colonnes du champ. La valeur -1 définit la largeur des colonnes sur Auto. Syntaxe : NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	Définit la justification des colonnes du champ. Syntaxe : NODE.justify.FIELDNAME

Propriétés de transformnode



Le noeud Transformation vous permet de sélectionner et de prévisualiser les résultats des transformations avant de les appliquer aux champs sélectionnés. [Pour plus d'informations, reportez-vous à la section Noeud Transformation dans le chapitre 6 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create transformnode
set :transformnode.fields = [AGE INCOME]
set :transformnode.formula = Select
set :transformnode.formula_log_n = true
set :transformnode.formula_log_n_offset = 1
```

propriétés transformnode	Le type de données	Description de la propriété
fields	[<i>champ1 ... champn</i>]	Champs à utiliser dans la transformation.
formula	All Select	Indique si toutes les transformations ou seules les transformations sélectionnées doivent être calculées.
formula_inverse	Booléen	Indique si la transformation inverse doit être utilisée.
formula_inverse_offset	nombre	Indique un décalage de données à utiliser pour la formule. Élément paramétré sur 0 par défaut, sauf s'il est défini par l'utilisateur.
formula_log_n	Booléen	Indique si la transformation par le logarithme _n doit être utilisée.
formula_log_n_offset	nombre	
formula_log_10	Booléen	Indique si la transformation par le logarithme ₁₀ doit être utilisée.
formula_log_10_offset	nombre	
formula_exponential	Booléen	Indique si la transformation exponentielle (e ^x) doit être utilisée.

propriétés transformnode	Le type de données	Description de la propriété
formula_square_root	<i>Booléen</i>	Indique si la transformation par la racine carrée doit être utilisée.
use_output_name	<i>Booléen</i>	Indique si un nom de sortie personnalisé est utilisé.
output_name	<i>chaîne</i>	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
paginate_output	<i>Booléen</i>	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	<i>nombre</i>	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
full_filename	<i>chaîne</i>	Indique le nom du fichier à utiliser pour la sortie de fichier.

Propriétés du nœud d'exportation

Propriétés communes des nœuds Exportation

Les propriétés suivantes sont communes à tous les nœuds d'exportation :

Propriété	Valeurs	Description de la propriété
publish_path	chaîne	Saisissez le nom racine à utiliser pour les fichiers d'images et de paramètres publiés.
publish_metadata	booléen	Spécifie si un fichier de métadonnées décrit les entrées et les sorties de l'image et de leurs modèles de données.
publish_use_parameters	booléen	Spécifie si des paramètres de flux sont inclus dans le fichier *.par.
publish_parameters	liste de chaîne	Spécifie les paramètres à inclure.
execute_mode	export_data publish	Spécifie si le nœud s'exécute sans publier le flux, aussi le flux est automatiquement publié lorsque le nœud est exécuté.

Propriétés cognosexportnode



Le nœud IBM Cognos BI Export exporte des données dans un format qui peut être lu par les bases de données Cognos BI. [Pour plus d'informations, reportez-vous à la section Nœud Export IBM Cognos BI dans le chapitre 7 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Remarque : Pour ce nœud, vous devez définir une connexion Cognos et une connexion ODBC.

Connexion Cognos

Les propriétés de la connexion Cognos sont les suivantes.

propriétés cognosexportnode	Le type de données	Description de la propriété
cognos_connection	{ "field", "field", ... ,"field" }	Une propriété de liste contenant les détails de la connexion du serveur Cognos. Le format est le suivant : { "URL_Serveur_Cognos", Mode_connexion, "espace de nommage", "nom utilisateur", "mot de passe" } où : URL_Serveur_Cognos est l'URL du serveur Cognos vers lequel vous effectuez l'exportation mode_connexion indique si la connexion anonyme est utilisée et est soit true soit

propriétés cognosexportnode	Le type de données	Description de la propriété
		false; si les champs suivants sont définis sur true, ils doivent être définis sur "" <i>espace de nommage</i> spécifie le fournisseur de sécurité pour l'authentification utilisé pour se connecter au serveur. <i>nom utilisateur</i> et <i>mot de passe</i> sont ceux utilisés pour la connexion au serveur Cognos
cognos_package_name	chaîne	Le chemin d'accès et le nom du package Cognos vers lequel vous exportez les données, par exemple : /Public Folders/MyPackage
cognos_datasource	chaîne	
cognos_export_mode	Publish ExportFile	
cognos_filename	chaîne	

connexion ODBC

Les propriétés de la connexion ODBC sont identiques à celles répertoriées pour `databaseexportnode` dans la prochaine section, à l'exception de la propriété `datasource` qui n'est pas valide.

Propriétés de `databaseexportnode`



Le noeud Export SGBD écrit des données dans une source de données relationnelles compatible ODBC. Pour que cette opération puisse être effectuée, la source de données ODBC doit exister et vous devez y avoir accès en écriture. [Pour plus d'informations](#), reportez-vous à la section [Noeud Export SGBD](#) dans le chapitre 7 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.

Exemple

```

/*
Use this sample with fraud.str from demo folder
Assumes a datasource named "MyDatasource" has been configured
*/
create databaseexport
connect claimvalue:applyneuralnetwork to :databaseexport
# Export tab
set :databaseexport.username = "user"
set :databaseexport.datasource = "MyDatasource"
set :databaseexport.password = "password"
set :databaseexport.table_name = "predictions"
set :databaseexport.write_mode = Create
set :databaseexport.generate_import = true
set :databaseexport.drop_existing_table = true
set :databaseexport.delete_existing_rows = true
set :databaseexport.default_string_size = 32

```



```
# Schema dialog
set :databaseexport.type.region = "VARCHAR(10)"
set :databaseexport.export_db_primarykey.id = true
set :databaseexportnode.use_custom_create_table_command = true
set :databaseexportnode.custom_create_table_command = "My SQL Code"

# Indexes dialog
set :databaseexport.use_custom_create_index_command = true
set :databaseexport.custom_create_index_command = \
  "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"
set :databaseexport.indexes.MYINDEX.fields = [id region]
```

propriétés databaseexportnode	Le type de données	Description de la propriété
datasource	chaîne	
username	chaîne	
password	chaîne	
epassword	chaîne	Cette propriété est en lecture seule au cours de l'exécution. Pour générer un mot de passe codé, sélectionnez Outil pour mot de passe dans le menu Outils. Pour plus d'informations, reportez-vous à la section Génération d'un mot de passe codé dans le chapitre 5 sur p. 62.
table_name	chaîne	
write_mode	Create Append Merge	
map	chaîne	Mappe un nom de champ de flux sur un nom de colonne de base de données (valide uniquement si write_mode est sur Merge). Exemple : set :databaseexportnode.map.streamBP = 'databaseBP' Le mappage multiple est pris en charge selon la position du champ, par exemple : set :databaseexportnode.map={{streamfield1 field1}{streamfield2 field2}{streamfield3 field3}} Pour une fusion, tous les champs doivent être mappés afin d'être exportés. Les noms de champ qui n'existent pas dans la base de données sont ajoutés en tant que nouvelles colonnes.

propriétés databaseexportnode	Le type de données	Description de la propriété
key_fields	<i>[champ champ ... champ]</i>	Spécifie le champ de flux utilisé pour la clé ; la propriété map indique à quoi cela correspond dans la base de données.
join	Database Add	Exemple : set :databaseexportnode.join = Database
drop_existing_table	<i>booléen</i>	
delete_existing_rows	<i>booléen</i>	
default_string_size	<i>entier</i>	
type		Propriété structurée utilisée pour définir le type de schéma. Syntaxe : set :databaseexportnode.type.BP = 'VARCHAR(10)'
generate_import	<i>booléen</i>	
use_custom_create_table_command	<i>booléen</i>	Utilisez la propriété <i>custom_create_table</i> pour modifier la commande SQL standard CREATE TABLE.
custom_create_table_command	<i>chaîne</i>	Indique la commande de chaîne à utiliser à la place de la commande SQL standard CREATE TABLE.
use_batch	<i>booléen</i>	Les propriétés suivantes sont les options avancées du chargement en masse dans la base de données. La valeur True (vrai) pour Use_batch désactive les validations ligne par ligne soumises à la base de données.
batch_size	<i>nombre</i>	Indique le nombre d'enregistrements à envoyer à la base de données avant validation dans la mémoire.
bulk_loading	Off ODBC External	Indique le type de chargement en masse. Les options supplémentaires relatives à ODBC et à External sont répertoriées ci-dessous.
odbc_binding	Row Column	Indiquez le lien par ligne ou par colonne pour le chargement en masse via ODBC.
loader_delimit_mode	Tab Space Other	Indiquez le type de séparateur pour le chargement en masse via un programme externe. Sélectionnez Other avec la propriété loader_other_delimiter pour indiquer les séparateurs, tels que la virgule (,).
loader_other_delimiter	<i>chaîne</i>	

propriétés databaseexportnode	Le type de données	Description de la propriété
specify_data_file	<i>booléen</i>	Un booléen ayant la valeur True (vrai) active la propriété <code>data_file</code> ci-dessous, vous permettant d'indiquer le nom et le chemin du fichier où écrire lors du chargement en masse dans la base de données.
<code>data_file</code>	<i>chaîne</i>	
specify_loader_program	<i>booléen</i>	Un booléen ayant la valeur True (vrai) active la propriété <code>loader_program</code> ci-dessous, vous permettant d'indiquer le nom et l'emplacement d'un programme ou d'un script de module de chargement externe.
<code>loader_program</code>	<i>chaîne</i>	
gen_logfile	<i>booléen</i>	Un booléen ayant la valeur True (vrai) active la propriété <code>logfile_name</code> ci-dessous, vous permettant d'indiquer le nom d'un fichier se trouvant sur le serveur pour générer un journal des erreurs.
<code>logfile_name</code>	<i>chaîne</i>	
check_table_size	<i>booléen</i>	Un booléen ayant la valeur True (vrai) permet de vérifier les tables pour garantir que l'augmentation de la taille des tables de la base de données correspond au nombre de lignes exportées à partir de IBM® SPSS® Modeler.
loader_options	<i>chaîne</i>	Indiquez des arguments supplémentaires (comme <code>-comment</code> et <code>-specialdir</code>) pour le programme du module de chargement.
export_db_primarykey	<i>booléen</i>	Indique si un champ donné est une clé primaire.
use_custom_create_index_command	<i>booléen</i>	Si la valeur true (vrai) est utilisée, active le code SQL personnalisé pour tous les index.
<code>custom_create_index_command</code>	<i>chaîne</i>	Indique la commande SQL utilisée pour créer des index lorsque le code SQL personnalisé est activé. (Cette valeur peut être ignorée pour certains index comme indiqué ci-dessous.)
<code>indexes.INDEXNAME.fields</code>		Crée l'index indiqué si nécessaire et répertorie les noms de champ à inclure dans cet index.

propriétés databaseexportnode	Le type de données	Description de la propriété
indexes.INDEXNAME.use_custom_create_index_command	booléen	Permet d'activer/de désactiver le code SQL personnalisé pour un index spécifique.
indexes.INDEXNAME.custom_create_command		Indique le code SQL personnalisé utilisé pour l'index spécifié.
indexes.INDEXNAME.remove	booléen	Si la valeur true (vrai) est utilisée, supprime l'index spécifié de l'ensemble d'index.
table_space	chaîne	Spécifie l'espace Table qui sera créé.
use_partition	booléen	Spécifie que le champ de distribution par hachage sera utilisé.
partition_field	chaîne	Spécifie le contenu du champ de distribution par hachage.

Remarque : pour certaines bases de données, vous pouvez spécifier que les tables de base de données doivent être créées pour l'exportation avec compression (par exemple, l'équivalent de CREATE TABLE MYTABLE (...) COMPRESS YES; dans SQL). Les propriétés use_compression et compression_mode sont fournies pour prendre en charge cette fonctionnalité, comme suit.

propriétés databaseexportnode	Le type de données	Description de la propriété
use_compression	Booléen	Si défini sur true, crée des tables à exporter avec la compression.
compression_mode	Row Page	Définit le niveau de compression des bases de données SQL Server.
	Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low	Définit le niveau de compression des bases de données Oracle. Veuillez noter que les valeurs OLTP, Query_High, Query_Low, Archive_High et Archive_Low requièrent Oracle 11gR2 au minimum.

Exemple - SQL Server

```
var DBSource
set DBSource = get node TestCompressionSQL
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Page
```

```
execute DBSource
```

Exemple - Oracle 11gR1

```
var DBSource
set DBSource = get node TestCompressionOracle11gR1
set ^DBSource.use_compression = true
```

```
set ^DBSource.compression_mode = Direct_Load_Operations
```

```
execute DBSource
```

Exemple - Oracle 11gR2

```
var DBSource
set DBSource = get node TestCompressionOracle11gR2
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Basic
```

```
execute DBSource
```

Propriétés de datacollectionexportnode



Le nœud Export IBM® SPSS® Data Collection génère des données au format utilisé par les logiciels d'étude de marché Data Collection. Pour pouvoir utiliser ce nœud, vous devez avoir installé avant la bibliothèque de données Data Collection. [Pour plus d'informations, reportez-vous à la section Nœud d'exportation IBM SPSS Data Collection dans le chapitre 7 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create datacollectionexportnode
set :datacollectionexportnode.metadata_file = "c:\museums.mdd"
set :datacollectionexportnode.merge_metadata = Overwrite
set :datacollectionexportnode.casedata_file = "c:\museumdata.sav"
set :datacollectionexportnode.generate_import = true
set :datacollectionexportnode.enable_system_variables = true
```

propriétés datacollectionexportnode	Le type de données	Description de la propriété
metadata_file	chaîne	Nom du fichier de métadonnées à exporter.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	booléen	Spécifie si le fichier <i>.mdd</i> exporté devrait inclure les variables système Data Collection.
casedata_file	chaîne	Le nom du fichier <i>.sav</i> vers lequel les données d'observation sont exportées.
generate_import	booléen	

Propriétés de *excelexportnode*



Le noeud Export Excel génère une sortie de données au format Microsoft Excel (.xls). Si vous le souhaitez, vous pouvez choisir de lancer Excel automatiquement et d'ouvrir le fichier exporté lors de l'exécution du noeud. [Pour plus d'informations, reportez-vous à la section Noeud Export Excel dans le chapitre 7 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create excelexportnode
set :excelexportnode.full_filename = "C:/output/myexport.xls"
set :excelexportnode.excel_file_type = Excel2007
set :excelexportnode.inc_field_names = True
set :excelexportnode.inc_labels_as_cell_notes = False
set :excelexportnode.launch_application = True
set :excelexportnode.generate_import = True
```

propriétés <i>excelexportnode</i>	Le type de données	Description de la propriété
full_filename	chaîne	
excel_file_type	Excel2003 Excel2007	
export_mode	Create Append	
inc_field_names	booléen	Indique si les noms de champ doivent être inclus dans la première ligne de la feuille de calcul.
start_cell	chaîne	Spécifie la cellule de démarrage de l'export.
worksheet_name	chaîne	Nom de la feuille de calcul à écrire.
launch_application	booléen	Indique si Excel doit être appelé pour le fichier obtenu. Notez que le chemin de lancement d'Excel doit être spécifié dans la boîte de dialogue Programmes externes (menu Outils, Programmes externes).
generate_import	booléen	Indique si un noeud Import Excel doit être généré pour lire le fichier de données exporté.

Propriétés de *outputfilenode*



L'exportation à l'aide d'un fichier plat génère des données dans un fichier texte délimité. Elles peuvent ainsi être lues par d'autres logiciels d'analyse ou par des tableurs. [Pour plus d'informations, reportez-vous à la section noeud Export Fichier plat dans le chapitre 7 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```

create outputfile
set :outputfile.full_filename = "c:/output/flatfile_output.txt"
set :outputfile.write_mode = Append
set :outputfile.inc_field_names = False
set :outputfile.use_newline_after_records = False
set :outputfile.delimit_mode = Tab
set :outputfile.other_delimiter = ","
set :outputfile.quote_mode = Double
set :outputfile.other_quote = "*"
set :outputfile.decimal_symbol = Period
set :outputfile.generate_import = True

```

propriétés outputfilenode	Le type de données	Description de la propriété
full_filename	chaîne	Nom du fichier de sortie.
write_mode	Overwrite Append	
inc_field_names	booléen	
use_newline_after_records	booléen	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	booléen	
generate_import	booléen	
encoding	StreamDefault SystemDefault "UTF-8"	

Propriétés de sasexportnode

Le nœud Export SAS permet d'obtenir des données de sortie au format SAS afin qu'elles puissent être lues par SAS ou par un logiciel compatible. Trois formats de fichier SAS sont disponibles : SAS pour Windows/OS2, SAS pour UNIX ou SAS version 7/8. [Pour plus d'informations, reportez-vous à la section Nœud Export SAS dans le chapitre 7 dans Nœuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```

create sasexportnode
set :sasexportnode.full_filename = "c:/output/SAS_output.sas7bdat"
set :sasexportnode.format = SAS8
set :sasexportnode.export_names = NamesAndLabels

```

```
set :sasexportnode.generate_import = True
```

propriétés sasexportnode	Le type de données	Description de la propriété
format	Windows UNIX SAS7 SAS8	Champs étiquette de propriétés de variante.
full_filename	chaîne	
export_names	NamesAndLabels NamesAsLabels	Permet de mapper les noms de champ de IBM® SPSS® Modeler lors de l'exportation vers des noms de variable IBM® SPSS® Statistics ou SAS.
generate_import	booléen	

Propriétés de statisticsexportnode



Le noeud Exporter Statistics génère des données au format IBM® SPSS® Statistics.sav. Les fichiers .sav peuvent être lus par SPSS Statistics Base et d'autres produits. Ce format est également utilisé pour les fichiers cache IBM® SPSS® Modeler. [Pour plus d'informations, reportez-vous à la section Noeud Exporter Statistics dans le chapitre 8 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Les propriétés de ce noeud sont décrites dans [Propriétés de statisticsexportnode sur p. 318.](#)

Propriétés de xmlexportnode



Le noeud Export XML génère une sortie de données dans un fichier au format XML. Vous pouvez également créer un noeud source XML pour lire de nouveau les données exportées dans le flux. [Pour plus d'informations, reportez-vous à la section Noeud Export XML dans le chapitre 7 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create xmlexportnode
set :xmlexportnode.full_filename = "c:\export\data.xml"
set :xmlexportnode.map = [{"/catalog/book/genre" genre} {"/catalog/book/title" title}]
```

propriétés xmlexportnode	Le type de données	Description de la propriété
full_filename	chaîne	(requis) Chemin complet et nom de fichier du fichier d'exportation XML.
use_xml_schema	booléen	Spécifiez s'il faut utiliser un schéma XML (fichier XSD ou DTD) pour contrôler la structure des données exportées.
full_schema_filename	chaîne	Chemin complet et nom de fichier et du fichier XSD ou DTD à utiliser. Requis si use_xml_schema est défini sur true (vrai).

propriétés <code>xmlportnode</code>	Le type de données	Description de la propriété
<code>generate_import</code>	<i>booléen</i>	Génère un nœud source XML pour lire le fichier de données exporté dans le flux.
<code>records</code>	<i>chaîne</i>	Expression XPath qui indique la limite de l'enregistrement.
<code>map</code>	<i>chaîne</i>	Mappe le nom de fichier dans une structure XML. Exemple : <code>set :xmlportnode.map = [{"/top/node1" field1} {"/top/node2" field2}]</code> Ceci mappe le champ de flux <code>field1</code> sur l'élément XML <code>/top/node1</code> , etc.

Propriétés de noeuds IBM SPSS Statistics

Propriétés de *statisticsimportnode*



Le noeud Statistics lit les données du format de fichier *.sav* utilisé par IBM® SPSS® Statistics ainsi que des fichiers cache enregistrés dans IBM® SPSS® Modeler, qui utilisent le même format. [Pour plus d'informations, reportez-vous à la section Noeud Statistics dans le chapitre 8 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create statisticsimportnode
set :statisticsimportnode.full_filename = "C:/data/drug1n.sav"
set :statisticsimportnode.import_names = true
set :statisticsimportnode.import_data = true
```

Propriétés de <i>statisticsimportnode</i>	Le type de données	Description de la propriété
full_filename	chaîne	Nom du fichier complet, y compris son chemin d'accès.
import_names	NamesAndLabels LabelsAsNames	Méthode de gestion des noms de variable et des étiquettes.
import_data	DataAndLabels LabelsAsData	Méthode de gestion des valeurs et des étiquettes.
use_field_format_for_storage	Booléen	Spécifie s'il convient d'utiliser les informations de format de champ SPSS Statistics lors de l'importation.

Propriétés *statistictransformnode*



Le noeud Transformation exécute une sélection de commandes de syntaxe IBM® SPSS® Statistics en fonction des sources de données dans IBM® SPSS® Modeler. Ce noeud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Noeud Transformation Statistics dans le chapitre 8 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create statistictransformnode
set :statistictransformnode.syntax = "COMPUTE NewVar = Na + K."
set :statistictransformnode.new_name.NewVar = "Mélange de médicaments"
```

```
set :statisticstransformnode.check_before_saving = true
```

propriétés statisticstransformnode	Le type de données	Description de la propriété
syntax	chaîne	
check_before_saving	Booléen	Valide la syntaxe saisie avant d'enregistrer les entrées. Affiche un message d'erreur si la syntaxe n'est pas valide.
default_include	Booléen	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.
include	Booléen	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.
new_name	chaîne	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.

Propriétés statisticsmodelnode



Le noeud Modèle Statistics vous permet d'analyser et de travailler avec vos données en exécutant des procédures IBM® SPSS® Statistics qui produisent un PMML. Ce noeud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Noeud Modèle Statistics dans le chapitre 8 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Exemple

```
create statisticsmodelnode
set :statisticsmodelnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticsmodelnode.new_name.NewVar = "Mixed Drugs"
```

propriétés statisticsmodelnode	Le type de données	Description de la propriété
syntax	chaîne	
default_include	Booléen	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.
include	Booléen	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.
new_name	chaîne	Pour plus d'informations, reportez-vous à la section Propriétés de filternode dans le chapitre 14 sur p. 163.

Propriétés de *statisticsoutputnode*



Le noeud Sortie Statistics vous permet d'appeler une procédure IBM® SPSS® Statistics pour analyser les données IBM® SPSS® Modeler. De nombreuses procédures d'analyses SPSS Statistics sont disponibles. Ce noeud requiert une copie avec licence de SPSS Statistics. [Pour plus d'informations, reportez-vous à la section Noeud Sortie Statistics dans le chapitre 8 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create statisticsoutputnode
set :statisticsoutputnode.syntax = "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A)"
set :statisticsoutputnode.use_output_name = False
set :statisticsoutputnode.output_mode = File
set :statisticsoutputnode.full_filename = "Observations par Age, Sexe et Antécédents médicaux"
set :statisticsoutputnode.file_type = HTML
```

Propriétés de <i>statisticsoutputnode</i>	Le type de données	Description de la propriété
mode	Dialog Syntax	Sélectionne l'option « Boîte de dialogue SPSS Statistics » ou l'Éditeur de syntaxe.
syntax	chaîne	
use_output_name	Booléen	
output_name	chaîne	
output_mode	Screen File	
full_filename	chaîne	
file_type	HTML SPV SPW	

Propriétés de *statisticexportnode*



Le noeud Exporter Statistics génère des données au format IBM® SPSS® Statistics.sav. Les fichiers .sav peuvent être lus par SPSS Statistics Base et d'autres produits. Ce format est également utilisé pour les fichiers cache IBM® SPSS® Modeler. [Pour plus d'informations, reportez-vous à la section Noeud Exporter Statistics dans le chapitre 8 dans Noeuds source, exécution et de sortie de IBM SPSS Modeler 15.](#)

Exemple

```
create statisticexportnode
set :statisticexportnode.full_filename = "c:/output/SPSS_Statistics_out.sav"
set :statisticexportnode.field_names = Names
set :statisticexportnode.launch_application = True
```

```
set :statisticsexportnode.generate_import = True
```

Propriétés de statisticsexportnode	Le type de données	Description de la propriété
full_filename	<i>chaîne</i>	
launch_application	<i>Booléen</i>	
export_names	NamesAndLabels NamesAsLabels	Permet de mapper les noms de champ de SPSS Modeler lors de l'exportation vers des noms de variable SPSS Statistics ou SAS.
generate_import	<i>Booléen</i>	

Propriétés du super noeud

Les propriétés propres aux super noeuds sont décrites dans les tableaux suivants : Remarque : les propriétés de noeud communes s'appliquent également aux super noeuds.

Table 22-1
source_supernode

Nom de la propriété	Type de propriété/Liste de valeurs	Description de la propriété
parameters	<i>tout</i>	Cette propriété permet de créer des paramètres et d'accéder à ceux spécifiés dans le tableau des paramètres d'un super noeud. Reportez-vous aux détails ci-dessous.

Table 22-2
process_supernode

Nom de la propriété	Type de propriété/Liste de valeurs	Description de la propriété
parameters	<i>tout</i>	Cette propriété permet de créer des paramètres et d'accéder à ceux spécifiés dans le tableau des paramètres d'un super noeud. Reportez-vous aux détails ci-dessous.

Table 22-3
terminal_supernode

Nom de la propriété	Type de propriété/Liste de valeurs	Description de la propriété
parameters	<i>tout</i>	Cette propriété permet de créer des paramètres et d'accéder à ceux spécifiés dans le tableau des paramètres d'un super noeud. Reportez-vous aux détails ci-dessous.
execute_method	Script Normal	
script	<i>chaîne</i>	

Paramètres du super noeud

Vous pouvez utiliser des scripts pour créer ou définir les paramètres du super noeud en utilisant le format général :

```
set mySuperNode.parameters.minvalue = 30
```

Vous pouvez également indiquer le type de super noeud en plus (ou à la place) du nom :

```
set :process_supernode.parameters.minvalue = 30
```

```
set mySuperNode:process_supernode.parameters.minvalue = 30
```

Vous pouvez également définir la valeur du paramètre à l'aide d'une expression CLEM :

```
set :process_supernode.parameters.minvalue = "<expression>"
```

Définition des propriétés des noeuds encapsulés

Vous pouvez paramétrer les propriétés de noeuds encapsulés spécifiques dans un super noeud en créant un paramètre de super noeud correspondant au nom littéral du noeud et à la propriété que vous souhaitez paramétrer. Par exemple, supposons que vous ayez un super noeud source avec un noeud Délimité encapsulé pour lire les données. Vous pouvez transmettre le nom du fichier à lire (indiqué par la propriété `full_filename`) comme suit :

```
set :source_supernode.parameters.':variablefilenode.full_filename' = "c:/mydata.txt"
```

Cela crée un paramètre de super noeud nommé `:variablefilenode.full_filename` avec une valeur de `c:/mydata.txt`. En supposant qu'un noeud du type indiqué existe dans le super noeud, sa valeur pour la propriété nommée sera définie en conséquence. Notez que cette opération est effectuée dans le script de flux, c'est-à-dire le script pour le flux qui *inclut* le super noeud, plutôt que le script du super noeud. Veillez à utiliser des guillemets simples pour indiquer le nom du paramètre.

Cette approche peut être utilisée avec n'importe quel noeud encapsulé, tant qu'il en résulte un noeud et un paramètre de propriété valides. Par exemple, pour paramétrer la propriété `rand_pct` pour un noeud Echantillon encapsulé, vous pouvez utiliser l'un des éléments suivants :

```
set mySuperNode.parameters.':samplenode.rand_pct' = 50
```

ou

```
set mySuperNode.parameters.'Sample.rand_pct'= 50
```

ou

```
set mySuperNode.parameters.'Sample:samplenode.rand_pct'= 50
```

Le premier paramètre ci-dessus suppose que le flux ne comporte qu'un seul noeud Echantillon ; le deuxième, qu'il ne comporte qu'un seul noeud nommé « Echantillon », quel qu'en soit le type. Le troisième paramètre est le plus explicite car il indique à la fois le nom et le type du noeud.

[Pour plus d'informations, reportez-vous à la section Paramètres du super noeud dans le chapitre 9 dans *Noeuds source, exécution et de sortie de IBM SPSS Modeler 15*.](#)

Limites des scripts de super noeud. Les super noeuds ne peuvent pas manipuler d'autres flux, ni modifier le flux actuel. Par conséquent, les commandes qui concernent les flux, comme `open stream`, `get stream`, `execute_script`, etc., ne peuvent pas être utilisées dans les scripts de super noeud.

Remarques

Ces informations ont été développées pour les produits et services offerts dans le monde.

Il est possible qu'IBM n'offre pas dans les autres pays les produits, services et fonctionnalités décrits dans ce document. Contactez votre représentant local IBM pour obtenir des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, programme ou service IBM n'implique pas que les seuls les produits, programmes ou services IBM peuvent être utilisés. Tout produit, programme ou service de fonctionnalité équivalente qui ne viole pas la propriété intellectuelle IBM peut être utilisé à la place. Cependant l'utilisateur doit évaluer et vérifier l'utilisation d'un produit, programme ou service non IBM.

IBM peut posséder des brevets ou des applications de brevet en attente qui couvrent les sujets décrits dans ce document. L'octroi de ce document n'équivaut aucunement à celui d'une licence pour ces brevets. Vous pouvez envoyer par écrit des questions concernant la licence à :

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, États-Unis

Pour obtenir des informations de licence concernant la configuration de caractères codés sur deux octets (DBCS), veuillez contacter dans votre pays le département chargé de la propriété intellectuelle chez IBM ou envoyez vos commentaires par écrit à :

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japon.

Le paragraphe suivant ne s'applique pas au Royaume-Uni ni à aucun pays dans lequel ces dispositions sont contraires au droit local : INTERNATIONAL BUSINESS MACHINES FOURNIT CETTE PUBLICATION « EN L'ÉTAT » SANS GARANTIE D'AUCUNE SORTE, IMPLICITE OU EXPLICITE, Y COMPRIS, MAIS SANS ETRE LIMITE AUX GARANTIES IMPLICITES DE NON VIOLATION, DE QUALITE MARCHANDE OU D'ADAPTATION POUR UN USAGE PARTICULIER. Certains états n'autorisent pas l'exclusion de garanties explicites ou implicites lors de certaines transactions, par conséquent, il est possible que cet énoncé ne vous concerne pas.

Ces informations peuvent contenir des erreurs techniques ou des erreurs typographiques. Ces informations sont modifiées de temps en temps ; ces modifications seront intégrées aux nouvelles versions de la publication. IBM peut apporter des améliorations et/ou modifications des produits et/ou des programmes décrits dans cette publications à tout moment sans avertissement préalable.

Toute référence dans ces informations à des sites Web autres qu'IBM est fournie dans un but pratique uniquement et ne sert en aucun cas de recommandation pour ces sites Web. Le matériel contenu sur ces sites Web ne fait pas partie du matériel de ce produit IBM et l'utilisation de ces sites Web se fait à vos propres risques.

IBM peut utiliser ou distribuer les informations que vous lui fournissez, de la façon dont il le souhaite, sans encourir aucune obligation envers vous.

Les personnes disposant d'une licence pour ce programme et qui souhaitent obtenir des informations sur celui-ci pour activer : (i) l'échange d'informations entre des programmes créés de manière indépendante et d'autres programmes (notamment celui-ci) et (ii) l'utilisation mutuelle des informations qui ont été échangées, doivent contacter :

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, États-Unis.

Ces informations peuvent être disponibles, soumises à des conditions générales, et dans certains cas payantes.

Le programme sous licence décrit dans ce document et toute la documentation sous licence disponible pour ce programme sont fournis par IBM en conformité avec les conditions de l'accord du client IBM, avec l'accord de licence du programme international IBM et avec tout accord équivalent entre nous.

Toutes les données sur les performances contenues dans le présent document ont été obtenues dans un environnement contrôlé. Par conséquent, les résultats obtenus dans d'autres environnements d'exploitation peuvent varier de manière significative. Certaines mesures peuvent avoir été effectuées sur des systèmes en cours de développement et il est impossible de garantir que ces mesures seront les mêmes sur les systèmes commercialisés. De plus, certaines mesures peuvent avoir été estimées par extrapolation. Les résultats réels peuvent être différents. Les utilisateurs de ce document doivent vérifier les données applicables à leur environnement spécifique.

Les informations concernant les produits autres qu'IBM ont été obtenues auprès des fabricants de ces produits, leurs annonces publiques ou d'autres sources publiques disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances, leur compatibilité ou toute autre fonctionnalité associée à des produits autres qu'IBM. Les questions sur les capacités de produits autres qu'IBM doivent être adressées aux fabricants de ces produits.

Toutes les déclarations concernant la direction ou les intentions futures d'IBM peuvent être modifiées ou retirées sans avertissement préalable et représentent uniquement des buts et des objectifs.

Ces informations contiennent des exemples de données et de rapports utilisés au cours d'opérations quotidiennes standard. Pour les illustrer le mieux possible, ces exemples contiennent des noms d'individus, d'entreprises, de marques et de produits. Tous ces noms sont fictifs et toute ressemblance avec des noms et des adresses utilisés par une entreprise réelle ne serait que pure coïncidence.

Si vous consultez la version papier de ces informations, il est possible que certaines photographies et illustrations en couleurs n'apparaissent pas.

Marques commerciales

IBM, le logo IBM, ibm.com et SPSS sont des marques commerciales d'IBM Corporation, déposées dans de nombreuses juridictions du monde entier. Une liste à jour des marques IBM est disponible sur Internet à l'adresse <http://www.ibm.com/legal/copytrade.shtml>.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques commerciales ou des marques déposées de Intel Corporation ou de ses filiales aux États-Unis et dans d'autres pays.

Linux est une marque déposée de Linus Torvalds aux États-Unis et/ou dans d'autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques commerciales de Microsoft Corporation aux Etats-Unis et/ou dans d'autres pays.

UNIX est une marque déposée de The Open Group aux Etats-Unis et dans d'autres pays.

Java et toutes les marques et logos Java sont des marques commerciales de Sun Microsystems, Inc. aux Etats-Unis et/ou dans d'autres pays.

Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.



- activer la commande NODE, 40
- Affichage des modèles de séries temporelles ISW IBM
 - propriétés de génération de scripts de nœud, 281
- Annotations
 - accès aux scripts, 69
- Arbre décision MS
 - propriétés de génération de scripts de nœud, 266, 269
- arguments
 - connexion au serveur, 73
 - fichier de commande, 72
 - IBM SPSS Collaboration and Deployment Services Repository connexion, 74
 - système, 75
- blanc
 - suppression dans les chaînes, 94
- boucles
 - utilisation dans des scripts, 59, 67–68
- caractère barre oblique inversée dans les expressions
 - CLEM, 79
- caractères, 77, 79
- chaînes, 77, 79
 - changement d'observation, 59
 - génération de scripts, 21
- chaînes littérales
 - imbrication dans les scripts, 30
- chaînes texte
 - imbrication dans les scripts, 30
- champs, 77, 79
 - désactivation lors de la génération de scripts, 182
- champs booléens
 - propriété des valeurs, 67
- champs continus
 - propriété des valeurs, 67
- champs nominaux
 - propriété des valeurs, 67
- champs Temps
 - conversion , 103
- CLEM
 - expressions, 77
 - génération de scripts, 9, 20
 - Langage, 77
 - types de données, 78–79
- CLEM expressions
 - paramètres, 26
- CLEM fonctions
 - aléatoire, 94
 - chaîne, 94
 - comparaison, 87
 - fonctions spéciales, 112
 - global, 110
 - informations, 85
 - logique, 89
 - numérique, 90
 - probabilité, 92
 - séquence, 104, 106
 - sur les bits, 92
 - trigonométrique, 91
 - valeurdate, 100
 - valeurs nulles et non renseignées, 111
- commande clear generated palette, 47, 63
- commande clear stream, 51
- commande close FILE, 55
- commande close STREAM, 50
- commande connect NODE, 39
- commande create NODE, 38
- commande create stream, 50
- commande de définition globale, 116
- commande de fermeture de programme, 28, 32
- commande delete model, 47
- commande delete NODE, 40
- commande delete output, 57
- commande disconnect NODE, 40
- commande duplicate NODE, 40
- commande execute NODE, 41
- commande execute_all, 32
- commande execute_project, 53
- commande execute_script, 32
- commande export model, 47
- commande export NODE, 41
- commande export output, 57
- commande flush NODE, 42
- commande for, 25, 28, 59, 65, 68
- commande for...endfor, 33
- commande get, 23
- commande get node, 42
- commande get output, 57
- commande get stream, 51
- commande if, 28, 65
- commande if...then...else, 34
- commande insert model, 49
- commande load model, 49
- commande load node, 42
- commande load output, 58
- commande load project, 54
- commande load state, 54
- commande load stream, 51
- commande open FILE, 55
- commande open stream, 25, 51
- commande position NODE, 42
- commande rename NODE, 25, 43
- commande retrieve, 60
- commande retrieve model, 49
- commande retrieve node, 43
- commande retrieve output, 58
- commande retrieve project, 54
- commande retrieve stream, 51
- commande save, 23
- commande save model, 49
- commande save node, 44

- commande save output, 58
- commande save project, 54
- commande save STREAM, 52
- commande set, 21, 25–26, 34
- commande store, 60
- commande store model, 49
- commande store node, 44
- commande store output, 58
- commande store project, 54
- commande store stream, 52
- commande value, 55
- commande var, 21, 25, 38
- commande with stream, 25, 53
- commande write FILE, 56
- commande writeln FILE, 56, 65, 68
- commentaires
 - génération de scripts, 29
- commutateurs
 - arguments de ligne de commande, 71
 - combinaison de plusieurs commutateurs, 72
- concaténation de chaînes, 86
- conventions, 85

- dates
 - conversion, 103
 - Manipulation, 103
- désactiver la commande NODE, 40
- détection d'erreurs
 - génération de scripts, 62
- directives d'accroissement d'arbre
 - imbriication dans les scripts, 30
- distribution Chi-deux
 - fonctions de probabilité, 92
- distribution F
 - fonctions de probabilité, 92
- Distribution normale
 - fonctions de probabilité, 92
- distribution t
 - fonctions de probabilité, 92
- documentation, 4

- entiers, 77–78
- espaces
 - suppression dans les chaînes, 94
- exécution de scripts, 16
- exemples
 - Aperçu, 6
- Exemples
 - Guide des applications, 4
- exemples d'application, 4
- exportation
 - Modèles, 47
 - noeuds, 41
 - PMML, 41, 47
 - SQL, 41, 47
- expressions, 77

- expressions CLEM
 - génération de scripts, 29, 34
 - recherche et remplacement de texte, 17

- flux
 - commande de définition globale, 115
 - génération de scripts, 9–10
 - propriétés, 120
- fonction @BLANK, 85, 111
- fonction @DIFF, 104, 106
- fonction (exponentielle) de puissance, 90
- fonction @FIELD, 112
- fonction @FIELDS_BETWEEN, 112
- fonction @FIELDS_MATCHING, 112
- fonction @INDEX, 104, 106
- fonction @LAST_NON_BLANK, 104, 106, 111
- fonction @MAX, 104, 106
- fonction @MEAN, 104, 106
- fonction @MIN, 104, 106
- fonction @MULTI_RESPONSE_SET, 112
- fonction @NULL, 85, 111
- fonction @OFFSET, 104, 106
- fonction @PARTITION_FIELD, 112
- fonction @PREDICTED, 112
- fonction @SDEV, 104, 106
- fonction @SINCE, 104, 106
- fonction @SUM, 104, 106
- fonction @TARGET, 112
- fonction @TESTING_PARTITION, 112
- fonction @THIS, 104, 106
- fonction @TODAY, 100
- fonction @TRAINING_PARTITION, 112
- fonction @VALIDATION_PARTITION, 112
- fonction abs, 90
- fonction aléatoire, 94
- fonction allbutfirst, 94
- fonction allbutlast, 94
- fonction alphabefore, 94
- fonction arccos, 91
- fonction arccosh, 91
- fonction arcsin, 91
- fonction arcsinh, 91
- fonction arctan, 91
- fonction arctan2, 91
- fonction arctanh, 91
- fonction cdf_chisq, 92
- fonction cdf_f, 92
- fonction cdf_normal, 92
- fonction cdf_t, 92
- fonction cos, 91
- fonction cosh, 91
- fonction count_equal, 87
- fonction count_greater_than, 87
- fonction count_less_than, 87
- fonction count_non_nulls, 87
- fonction count_not_equal, 87
- fonction count_nulls, 87

-
- fonction count_substring, 94
 - fonction date_before, 87
 - fonction datetime_date, 86
 - fonction DIFF, 106
 - fonction div, 90
 - fonction endstring, 94
 - fonction exponentielle, 90
 - fonction first_index, 87
 - fonction first_non_null, 87
 - fonction first_non_null_index, 87
 - fonction fracof, 90
 - fonction hasendstring, 94
 - fonction hasmidstring, 94
 - fonction hasstartstring, 94
 - fonction hassubstring, 94
 - fonction INDEX, 106
 - fonction integer_bitcount, 92
 - fonction integer_leastbit, 92
 - fonction integer_length, 92
 - fonction intof, 90
 - fonction is_date, 85
 - fonction is_datetime, 85
 - fonction is_integer, 85
 - fonction is_number, 85
 - fonction is_real, 85
 - fonction is_string, 85
 - fonction is_time, 85
 - fonction is_timestamp, 85
 - fonction isalphacode, 94
 - fonction isendstring, 94
 - fonction islowercode, 94
 - fonction ismidstring, 94
 - fonction isnumbercode, 94
 - fonction isstartstring, 94
 - fonction issubstring, 94
 - fonction issubstring_count, 94
 - fonction issubstring_lim, 94
 - fonction isuppercode, 94
 - fonction last_index, 87
 - fonction LAST_NON_BLANK, 106
 - fonction last_non_null, 87
 - fonction last_non_null_index, 87
 - fonction length, 94
 - fonction locchar, 94
 - fonction locchar_back, 94
 - fonction log, 90
 - fonction log10, 90
 - fonction lowertoupper, 59, 94
 - fonction matches, 94
 - fonction max, 87
 - fonction MAX, 106
 - fonction max_index, 87
 - fonction max_n, 87
 - fonction MEAN, 104, 106
 - fonction mean_n, 90
 - fonction member, 87
 - fonction min, 87
 - fonction MIN, 106
 - fonction min_index, 87
 - fonction min_n, 87
 - fonction mod, 90
 - fonction negate, 90
 - fonction OFFSET, 106
 - fonction oneof, 94
 - fonction pi, 91
 - fonction random0, 94
 - fonction rem, 90
 - fonction replace, 94
 - fonction replicate, 94
 - fonction round, 90
 - fonction SDEV, 106
 - fonction sdev_n, 90
 - fonction sign, 90
 - fonction sin, 91
 - fonction SINCE, 106
 - fonction sinh, 91
 - fonction skipchar, 94
 - fonction skipchar_back, 94
 - fonction soundex, 99
 - fonction soundex_difference, 99
 - fonction sqrt, 90
 - fonction startstring, 94
 - fonction stripchar, 94
 - fonction strmember, 94
 - fonction subscrs, 94
 - fonction substring, 94
 - fonction substring_between, 94
 - fonction SUM, 106
 - fonction sum_n, 90
 - fonction tan, 91
 - fonction tanh, 91
 - fonction testbit, 92
 - fonction THIS, 106
 - fonction time_before, 87
 - fonction to_date, 86, 100
 - fonction to_dateline, 100
 - fonction to_datetime, 86
 - fonction to_integer, 86
 - fonction to_number, 86
 - fonction to_real, 86
 - fonction to_string, 86
 - fonction to_time, 86, 100
 - fonction to_timestamp, 86, 100
 - fonction trim, 94
 - fonction trim_start, 94
 - fonction trimend, 94
 - fonction undef, 111
 - fonction unicode_char, 94
 - fonction unicode_value, 94
 - fonction uppertolower, 94
 - fonction value_at, 87
 - fonctions, 80–81, 85, 104
 - @FIELD, 112
 - @GLOBAL_MAX, 110

- @GLOBAL_MEAN, 110
- @GLOBAL_MIN, 110
- @GLOBAL_SDEV, 110
- @GLOBAL_SUM, 110
- @PARTITION, 112
- @PREDICTED, 112
- @TARGET, 112
- fonctions CLEM
 - Conversion, 86
 - liste des fonctions disponibles, 83
- fonctions datetime
 - datetime_date, 100
 - datetime_day, 100
 - datetime_day_name, 100
 - datetime_day_short_name, 100
 - datetime_hour, 100
 - datetime_in_seconds, 100
 - datetime_minute, 100
 - datetime_month, 100
 - datetime_month_name, 100
 - datetime_month_short_name, 100
 - datetime_now datetime_second, 100
 - datetime_time, 100
 - datetime_timestamp, 100
 - datetime_weekday, 100
 - datetime_year, 100
- fonctions de comparaison, 87
- fonctions de conversion, 86
- fonctions de date, 80–81
 - date_before, 87, 100
 - date_days_difference, 100
 - date_in_days, 100
 - date_in_months, 100
 - date_in_weeks, 100
 - date_in_years, 100
 - date_months_difference, 100
 - date_weeks_difference, 100
 - date_years_difference, 100
 - fonction @TODAY, 100
- Fonctions de distribution, 92
- fonctions de probabilité, 92
- fonctions d'heure, 80–81
 - time_before, 87, 100
 - time_hours_difference, 100
 - time_in_hours, 100
 - time_in_mins, 100
 - time_in_secs, 100
 - time_mins_difference, 100
 - time_secs_difference, 100
- fonctions d'heure et de date, 80–81
- fonctions d'informations, 85
- fonctions globales, 110
- fonctions if, then, else, 89
- Fonctions logiques, 89
- fonctions numériques, 90
- fonctions séquentielles, 104, 106
- fonctions spéciales, 112
- Fonctions sur chaînes, 59, 94
- fonctions sur les bits, 92
- fonctions trigonométriques, 91
- Format HTML
 - exportation de modèles, 47
 - exportation de noeuds, 41
- format PMML
 - exportation de modèles, 47
 - exportation de noeuds, 41
- Format SQL
 - exportation de noeuds, 41, 47
- Format texte
 - exportation de modèles, 47
 - exportation de noeuds, 41
- Formats de date, 80–81
- formats d'heure, 80–81
- Générateur de formules
 - recherche et remplacement de texte, 17
- génération de scripts
 - à partir de la ligne de commande, 63
 - abréviations utilisées, 117
 - Aperçu, 9, 20
 - commentaires, 29
 - compatibilité avec les versions antérieures, 63
 - dans les super noeuds, 15
 - détection d'erreurs, 62
 - exécution, 16
 - exécution de scripts, 28
 - Exemples, 65, 68
 - expressions CLEM, 29
 - flux, 9
 - interface utilisateur, 10, 12, 15
 - interruption, 16
 - lignes incomplètes, 29
 - modèles Sélection de fonction, 14
 - noeuds, 21
 - noeuds de sortie, 291
 - noeuds Graphiques, 182
 - objet actuel, 23
 - opérateurs, 28
 - ordre d'exécution de flux, 59
 - propriétés communes, 118
 - recherche et remplacement de texte, 17
 - scripts autonomes, 9
 - scripts de super noeud, 9
 - syntax, 20
- gestion des blancs
 - fonctions CLEM, 111
- IBM SPSS Collaboration and Deployment Services Repository
 - arguments de ligne de commande, 74
 - génération de scripts, 60
- IBM SPSS Modeler, 1
 - démarrage à partir de la ligne de commande, 71
 - documentation, 4

- id de noeud
 - référence dans les scripts, 21
- interruption de scripts, 16
- Introduction, 77
- ligne de commande
 - arguments multiples, 72
 - démarrage de IBM SPSS Modeler, 71
 - exécution de IBM SPSS Modeler, 71
 - génération de scripts, 63
 - liste des arguments, 73–75
 - paramètres, 76
- lignes incomplètes
 - génération de scripts, 29
- listes, 77, 79
- littéraux
 - génération de scripts, 20, 30
- marques commerciales, 323
- mentions légales, 322
- Modèles
 - exportation, 47
 - génération de scripts, 47
 - noms de scripts, 44, 46
- modèles ACP
 - propriétés de génération de scripts de noeud, 220, 260
- Modèles ACP Netezza
 - propriétés de génération de scripts de noeud, 286, 290
- Modèles ACP/Analyse factorielle
 - propriétés de génération de scripts de noeud, 220, 260
- modèles apriori
 - propriétés de génération de scripts de noeud, 199, 255
- Modèles Apriori Oracle
 - propriétés de génération de scripts de noeud, 274, 276
- Modèles Bayes Net Netezza
 - propriétés de génération de scripts de noeud, 285, 290
- Modèles C5.0
 - propriétés de génération de scripts de noeud, 208, 257
- Modèles CARMA
 - propriétés de génération de scripts de noeud, 209, 258
- Modèles CHAID
 - propriétés de génération de scripts de noeud, 213, 258
- modèles d'agrégation suivant le saut minimum
 - propriétés de génération de scripts de noeud, 231
- Modèles d'arbre C&RT
 - propriétés de génération de scripts de noeud, 210, 258
- Modèles d'arbre de décision ISW IBM
 - propriétés de génération de scripts de noeud, 277, 282
- Modèles d'arbre de décision Netezza
 - propriétés de génération de scripts de noeud, 283, 290
- Modèles d'arbre de régression Netezza
 - propriétés de génération de scripts de noeud, 286, 290
- Modèles d'arbre décision Oracle
 - propriétés de génération de scripts de noeud, 273, 276
- Modèles d'association ISW IBM
 - propriétés de génération de scripts de noeud, 277, 282
- Modèles de classificateur automatique
 - propriétés de génération de scripts de noeud, 256
- Modèles de classification automatique
 - propriétés de génération de scripts de noeud, 256
- Modèles de classification ISW IBM
 - propriétés de génération de scripts de noeud, 280, 282
- Modèles de classification par division Netezza
 - propriétés de génération de scripts de noeud, 286, 290
- modèles de détection des anomalies
 - propriétés de génération de scripts de noeud, 198, 255
- Modèles de nuées dynamiques
 - propriétés de génération de scripts de noeud, 230, 260
- modèles de numérisation automatique
 - propriétés de génération de scripts de noeud, 205
- Modèles de numérisation automatique
 - propriétés de génération de scripts de noeud, 257
- Modèles de régression de Cox
 - propriétés de génération de scripts de noeud, 215, 259
- Modèles de régression ISW IBM
 - propriétés de génération de scripts de noeud, 279, 282
- modèles de régression linéaire
 - propriétés de génération de scripts de noeud, 246, 263
- Modèles de régression linéaire Netezza
 - propriétés de génération de scripts de noeud, 287, 290
- modèles de régression logistique
 - propriétés de génération de scripts de noeud, 235, 261
- Modèles de régression logistique ISW IBM
 - propriétés de génération de scripts de noeud, 281–282
- Modèles de Réponse en auto-apprentissage
 - propriétés de génération de scripts de noeud, 249, 263
- modèles de réseau Bayésien
 - propriétés de génération de scripts de noeud, 206
- Modèles de réseau Bayésien
 - propriétés de génération de scripts de noeud, 257
- modèles de réseau de neurones
 - propriétés de génération de scripts de noeud, 240, 262
- Modèles de séquence ISW IBM
 - propriétés de génération de scripts de noeud, 278, 282
- modèles de séquences
 - propriétés de génération de scripts de noeud, 248, 264
- modèles de séries temporelles
 - propriétés de génération de scripts de noeud, 251, 264
- Modèles de séries temporelles Netezza
 - propriétés de génération de scripts de noeud, 287
- Modèles d'IA d'Oracle
 - propriétés de génération de scripts de noeud, 275
- modèles discriminants
 - propriétés de génération de scripts de noeud, 218, 259
- modèles générés
 - noms de scripts, 44, 46
- Modèles IBM DB2
 - propriétés de génération de scripts de noeud, 276
- modèles IBM SPSS Statistics
 - propriétés de génération de scripts de noeud, 317
- Modèles KMeans Oracle
 - propriétés de génération de scripts de noeud, 273, 276

- modèles KNN
 - propriétés de génération de scripts de noeud, 261
- Modèles KNN Netezza
 - propriétés de génération de scripts de noeud, 285, 290
- modèles Kohonen
 - propriétés de génération de scripts de noeud, 233
- Modèles Kohonen
 - propriétés de génération de scripts de noeud, 261
- modèles linéaires
 - propriétés de génération de scripts de noeud, 234, 261
- modèles linéaires généralisés
 - propriétés de génération de scripts de noeud, 224, 260
- Modèles linéaires généralisés d'Oracle
 - propriétés de génération de scripts de noeud, 272
- Modèles linéaires généralisés Netezza
 - propriétés de génération de scripts de noeud, 289
- modèles Liste de décision
 - propriétés de génération de scripts de noeud, 217, 259
- Modèles MDL Oracle
 - propriétés de génération de scripts de noeud, 274, 276
- Modèles Microsoft
 - propriétés de génération de scripts de noeud, 266, 268
- Modèles MMLG
 - propriétés de génération de scripts de noeud, 227
- Modèles Naive Bayes d'IBM ISW
 - propriétés de génération de scripts de noeud, 280, 282
- Modèles Naive Bayes Netezza
 - propriétés de génération de scripts de noeud, 285, 290
- Modèles Netezza
 - propriétés de génération de scripts de noeud, 282
- Modèles Netezza K-means
 - propriétés de génération de scripts de noeud, 284, 290
- Modèles NMF Oracle
 - propriétés de génération de scripts de noeud, 274, 276
- modèles Oracle
 - propriétés de génération de scripts de noeud, 270
- Modèles Oracle Adaptive Bayes
 - propriétés de génération de scripts de noeud, 271, 275
- Modèles Oracle Naive Bayes
 - propriétés de génération de scripts de noeud, 271, 275
- Modèles Oracle Support Vector Machines
 - propriétés de génération de scripts de noeud, 272, 276
- Modèles QUEST
 - propriétés de génération de scripts de noeud, 244, 263
- modèles sélection de fonction
 - propriétés de génération de scripts de noeud, 222, 260
- modèles Sélection de fonction
 - application, 14
 - génération de scripts, 14
- Modèles SLRM
 - propriétés de génération de scripts de noeud, 249, 263
- modèles support vector machine
 - propriétés de génération de scripts de noeud, 264
- Modèles Support vector machine
 - propriétés de génération de scripts de noeud, 250
- Modèles SVM
 - propriétés de génération de scripts de noeud, 250
- modèles TwoStep
 - propriétés de génération de scripts de noeud, 254, 264
- modélisation de bases de données, 265
- mot-clé generated, 63
- mots de passe
 - ajout aux scripts, 62
 - codés, 73
- mots de passe codés
 - ajout aux scripts, 62
- MS Sequence Clustering
 - propriétés de génération de scripts de noeud, 270
- noeud Agréger
 - propriétés, 143
- Noeud Agréger RFM
 - propriétés, 147
- noeud Ajouter
 - propriétés, 143
- noeud Analyse
 - propriétés, 291
- Nœud Analyse RFM
 - propriétés, 169
- Nœud Anonymiser
 - propriétés, 153
- Noeud Audit données
 - propriétés, 292
- noeud Binariser
 - propriétés, 170
- noeud Calculer
 - propriétés, 159
- Noeud Classificateur automatique
 - propriétés de génération de scripts de noeud, 201
- Noeud classification automatique
 - propriétés de génération de scripts de noeud, 203
- noeud Courbes
 - propriétés, 190
- Noeud de sortie IBM SPSS Statistics
 - propriétés, 318
- noeud Délimité
 - propriétés, 138
- noeud d'exportation IBM SPSS Data Collection
 - propriétés, 311
- noeud d'exportation IBM SPSS Statistics
 - propriétés, 318
- noeud Discrétiser
 - propriétés, 157
- noeud Distinguer
 - propriétés, 145
- noeud Echantillon
 - propriétés, 149
- Nœud Ensemble
 - propriétés, 161
- Nœud Enterprise View
 - propriétés, 133
- noeud Equilibrer
 - propriétés, 144

- noeud Evaluation
 - propriétés, 185
- nœud export Excel
 - propriétés, 312
- nœud Export SAS
 - propriétés, 313
- nœud Export SGBD
 - propriétés, 306
- Nœud Export XML
 - propriétés, 314
- nœud Fichier plat
 - propriétés, 312
- Nœud Filtrer
 - propriétés, 163
- nœud Fixe
 - propriétés, 134
- noeud Fusionner
 - propriétés, 146
- noeud Histogramme
 - propriétés, 189
- nœud Historiser
 - propriétés, 164
- noeud Intervalles de temps
 - propriétés, 171
- noeud Matrice
 - propriétés, 294
- noeud Moyennes
 - propriétés, 296
- noeud Nuage
 - propriétés, 191
- nœud Partitionner
 - propriétés, 165
- noeud Proportion
 - propriétés, 184
- noeud Rapport, 65, 68
 - propriétés, 297
- nœud Re-trier
 - propriétés, 167
- noeud Recoder
 - propriétés, 166
- noeud Relations
 - propriétés, 195
- noeud Relations orientées
 - propriétés, 195
- noeud Remplacer
 - propriétés, 162
- Noeud Représentation Graphique
 - propriétés, 187
- nœud Restructurer
 - propriétés, 168
- noeud Résumé
 - propriétés, 183
- noeud Sélectionner
 - propriétés, 151
- Nœud SGBD.
 - propriétés, 128
- Nœud source Excel
 - propriétés, 132
- Nœud source IBM Cognos BI
 - propriétés, 126
- nœud source IBM SPSS Data Collection
 - propriétés, 130
- nœud source IBM SPSS Statistics
 - propriétés, 316
- Nœud source SAS
 - propriétés, 136
- Nœud source XML
 - propriétés, 141
- noeud Statistiques
 - propriétés, 299
- noeud Table
 - propriétés, 300
- noeud Tracé horaire
 - propriétés, 193
- noeud Transformation
 - propriétés, 303
- noeud Transformation IBM SPSS Statistics
 - propriétés, 316
- nœud Transposer
 - propriétés, 176
- noeud Trier
 - propriétés, 151
- noeud Typer
 - propriétés, 176
- Nœud Utilisateur
 - propriétés, 137
- noeud V. globales
 - propriétés, 298
- noeuds
 - bouclage dans les scripts, 59
- noeuds de modélisation
 - propriétés de génération de scripts de noeud, 197
- noeuds de sortie
 - propriétés de génération de scripts, 291
- noeuds d'exportation
 - propriétés de génération de scripts de noeud, 305
- noeuds Graphiques
 - propriétés de génération de scripts, 182
- noeuds source
 - propriétés, 124
- nombres, 78
- noms de champ
 - changement d'observation, 59
- noms de flux
 - accès aux scripts, 69
- nuggets
 - propriétés de génération de scripts de noeud, 255
- nuggets de modèle
 - noms de scripts, 44, 46
 - propriétés de génération de scripts de noeud, 255
- O-Cluster Oracle
 - propriétés de génération de scripts de noeud, 273, 276

- objet actuel
 - référence dans les scripts, 23
- objets de fichier
 - commande de script, 55
- objets de flux
 - commande de script, 50
 - Ouverture, 24–25
 - référencement, 25
- objets de modèle
 - commande de script, 44
 - noms de scripts, 44, 46
- objets de noeud
 - commande de script, 38
 - génération de scripts, 21
- objets de résultat
 - commande de script, 55
- objets de sortie
 - commande de script, 56
 - noms de scripts, 56
- objets d'état
 - commande de script, 54
- opérateur and, 89
- opérateur Différent de, 87
- opérateur Egal à, 87
- opérateur Inférieur à, 87
- opérateur not, 89
- opérateur or, 89
- opérateur Supérieur à, 87
- opérateurs
 - génération de scripts, 28
 - jonction de chaînes, 69, 86
- ordre d'exécution
 - changement à l'aide de scripts, 59
- ordre d'exécution de flux
 - changement à l'aide de scripts, 59

- paramètres, 15, 34, 115–117, 120
 - flux, 26
 - génération de scripts, 20, 29
 - session, 26
 - Super noeuds, 320
- paramètres de flux, 26, 34
- paramètres de liste
 - modification dans les scripts, 28
- paramètres de propriété, 15, 34, 115, 118
- paramètres de session, 26, 34
- préparation automatique des données
 - propriétés, 154
- priorité, 81
- priorité des opérateurs, 81
- projets
 - propriétés, 123
- properties cognosimportnode, 126
- propriété column_count, 55
- propriété des valeurs, 67
- propriété row_count, 55
- propriété stream.nodes, 59
- propriétés, 34
 - flux, 120
 - génération de scripts, 115–118, 197, 255, 305
 - génération de scripts commune, 118
 - nœuds de modélisation de base de données, 265
 - noeuds Filtrer, 116
 - projets, 123
 - Super noeuds, 320
- Propriétés applyknnnode, 261
- propriétés applynetezadectreenode, 290
- propriétés autoclassifiernode, 201
- propriétés autodatapreprenode, 154
- propriétés de aggregatenode, 143
- propriétés de analysisnode, 291
- propriétés de anomalydetectionnode, 198
- propriétés de anonymizenode, 153
- propriétés de appendnode, 143
- propriétés de applyanomalydetectionnode, 255
- propriétés de applyapriorinode, 255
- propriétés de applyautoclassifiernode, 256
- propriétés de applyautoclusternode, 256
- propriétés de applyautonumericnode, 257
- propriétés de applybayesnetnode, 257
- propriétés de applyc50node, 257
- propriétés de applycarmanode, 258
- propriétés de applycartnode, 258
- propriétés de applychaidnode, 258
- propriétés de applycoxregnode, 259
- propriétés de applydb2imclusternode, 282
- propriétés de applydb2imlognode, 282
- propriétés de applydb2imnbnode, 282
- propriétés de applydb2imregnode, 282
- propriétés de applydb2imtreenode, 282
- propriétés de applydecisionlistnode, 259
- propriétés de applydiscriminantnode, 259
- propriétés de applyfactornode, 260
- propriétés de applyfeatureselectionnode, 260
- propriétés de applygeneralizedlinearnode, 260
- propriétés de applykmeansnode, 260
- propriétés de applykohonennode, 261
- Propriétés de applylinearmode, 261
- Propriétés de applylogregnode, 261
- propriétés de applymslogisticnode, 269
- propriétés de applymsneuralnetworknode, 269
- propriétés de applymsregressionnode, 269
- propriétés de applymssequenceclusternode, 270
- propriétés de applymstimeseriesnode, 269
- propriétés de applymstreenode, 269
- propriétés de applynetezزابayesnode, 290
- propriétés de applynetezzadivclusternode, 290
- propriétés de applynetezzakmeansnode, 290
- propriétés de applynetezzaknnnode, 290
- propriétés de applynetezzalineregressionnode, 290
- propriétés de applynetezzanaivebayesnode, 290
- propriétés de applynetezzapcanode, 290
- propriétés de applynetezzaregtreenode, 290
- propriétés de applyneuralnetnode, 262

- Propriétés de applyneuralnetworknode, 262
- propriétés de applyoraabnode, 275
- propriétés de applyoradecisiontreenode, 276
- propriétés de applyorakmeansnode, 276
- propriétés de applyoranbnode, 275
- propriétés de applyoranmfnode, 276
- propriétés de applyoraoclusternode, 276
- propriétés de applyorasvmnode, 276
- propriétés de applyquestnode, 263
- propriétés de applyselflearningnode, 263
- propriétés de applysequencenode, 264
- propriétés de applysvmnode, 264
- propriétés de applytimeseriesnode, 264
- propriétés de applytwestepnode, 264
- propriétés de apriorinode, 199
- propriétés de autonumericnode, 205
- propriétés de balancenode, 144
- propriétés de bayesnet, 206
- propriétés de binningnode, 157
- propriétés de c50node, 208
- propriétés de carmanode, 209
- propriétés de cartnode, 210
- propriétés de chaidnode, 213
- propriétés de collectionnode, 183
- propriétés de coxregnode, 215
- propriétés de dataauditnode, 292
- propriétés de databaseexportnode, 306
- propriétés de databenode, 128
- propriétés de datacollectionexportnode, 311
- propriétés de datacollectionimportnode, 130
- propriétés de db2imassocnode, 277
- propriétés de db2imclusternode, 280
- propriétés de db2imlognode, 281
- propriétés de db2imnbnode, 280
- propriétés de db2imregnode, 279
- propriétés de db2imsequencenode, 278
- propriétés de db2imtimeseriesnode, 281
- propriétés de db2imtreenode, 277
- propriétés de decisionlist, 217
- propriétés de derivenode, 159
- propriétés de directedwebnode, 195
- propriétés de discriminantnode, 218
- propriétés de distinctnode, 145
- propriétés de distributionnode, 184
- propriétés de ensemblenode, 161
- propriétés de evaluationnode, 185
- propriétés de evimportnode, 133
- Propriétés de excelexportnode, 312
- propriétés de excelimportnode, 132
- propriétés de factornode, 220
- propriétés de featureselectionnode, 14, 222
- propriétés de fillernode, 162
- propriétés de filternode, 163
- propriétés de fixedfilenode, 134
- propriétés de flatfilenode, 312
- propriétés de génération de scripts de nœud, 265
 - nœuds de modélisation, 197
 - nœuds d'exportation, 305
 - nuggets de modèle, 255
- propriétés de genlinnode, 224
- propriétés de graphboardnode, 187
- propriétés de histogramnode, 189
- propriétés de historynode, 164
- propriétés de kmeansnode, 230
- propriétés de kohonennode, 233
- propriétés de logregnode, 235
- propriétés de matrixnode, 294
- propriétés de meansnode, 296
- propriétés de mergenode, 146
- propriétés de msassocnode, 266
- propriétés de msbayesnode, 266
- propriétés de msclusternode, 266
- propriétés de mslogisticnode, 266
- propriétés de msneuralnetworknode, 266
- propriétés de msregressionnode, 266
- propriétés de mssequenceclusternode, 266
- propriétés de mstimeseriesnode, 266
- propriétés de mstreenode, 266
- propriétés de multiplotnode, 190
- propriétés de netezzabayesnode, 285
- propriétés de netezzadectreenode, 283
- propriétés de netezzadivclusternode, 286
- propriétés de netezzakmeansnode, 284
- propriétés de netezzaknnnode, 285
- propriétés de netezzanaivebayesnode, 285
- propriétés de netezzapcanode, 286
- propriétés de netezzareggreenode, 286
- propriétés de neuralnetnode, 240
- propriétés de neuralnetworknode, 242
- propriétés de nœud
 - accès aux scripts, 69
- propriétés de numericpredictornode, 205
- propriétés de oraabnode, 271
- propriétés de oraainode, 275
- propriétés de oraapriorinode, 274
- propriétés de oradecisiontreenode, 273
- propriétés de orakmeansnode, 273
- propriétés de oramdlnode, 274
- propriétés de oranbnode, 271
- propriétés de oranmfnode, 274
- propriétés de oraoclusternode, 273
- propriétés de orasvmnode, 272
- Propriétés de outputfilenode, 312
- propriétés de partitionnode, 165
- propriétés de plotnode, 191
- propriétés de questnode, 244
- propriétés de reclassifynode, 166
- propriétés de regressionnode, 246
- propriétés de reordernode, 167
- propriétés de reportnode, 297
- propriétés de restructurenode, 168
- propriétés de rfmaggregatenode, 147

- propriétés de rfmanalysisnode, 169
 - propriétés de samplenode, 149
 - propriétés de sasexportnode, 313
 - propriétés de sasimportnode, 136
 - propriétés de selectnode, 151
 - propriétés de sequencenode, 248
 - propriétés de setglobalsnode, 298
 - propriétés de settoflagnode, 170
 - propriétés de slrmnode, 249
 - propriétés de sortnode, 151
 - propriétés de statisticsexportnode, 318
 - propriétés de statisticsimportnode, 14, 316
 - propriétés de statisticsnode, 299
 - propriétés de statisticsoutputnode, 318
 - propriétés de svmnode, 250
 - propriétés de tablenode, 300
 - propriétés de timeintervalnode, 171
 - propriétés de timeplotnode, 193
 - propriétés de timeseriesnode, 251
 - propriétés de transformnode, 303
 - propriétés de transposenode, 176
 - propriétés de twostepnode, 254
 - propriétés de typenode, 14, 66, 176
 - propriétés de userinputnode, 137
 - propriétés de variablefilenode, 138
 - propriétés de webnode, 195
 - Propriétés de xmlexportnode, 314
 - propriétés de xmlimportnode, 141
 - propriétés du flux, 69
 - propriétés du noeud de classification automatique, 203
 - propriétés glmnode, 227
 - propriétés knnnode, 231
 - propriétés linéaires, 234
 - propriétés netezzaglmnode, 289
 - propriétés netezzalinerégressionnode, 287
 - propriétés netezzatimeseriesnode, 287
 - propriétés oraglmnode, 272
 - propriétés statisticsmodelnode, 317
 - propriétés statisticstransformnode, 316
 - propriétés structurées, 116
-
- recherche de texte, 17
 - réels, 77–78
 - Régression linéaire MS
 - propriétés de génération de scripts de noeud, 266, 269
 - Régression logistique MS
 - propriétés de génération de scripts de noeud, 266, 269
 - remplacement de texte, 17
 - Réseau neuronal MS
 - propriétés de génération de scripts de noeud, 266, 269
 - réseaux de neurones
 - propriétés de génération de scripts de noeud, 242, 262
-
- scripts
 - enregistrement, 10
 - importation à partir de fichiers texte, 10
 - scripts autonomes, 9, 12
-
- sécurité
 - mots de passe codés, 62, 73
 - Séries temporelles MS
 - propriétés de génération de scripts de noeud, 269
 - serveur
 - arguments de ligne de commande, 73
 - sortie HTML
 - création et utilisation de scripts, 65, 68
 - SPSS Modeler Server, 2
 - super noeud, 115
 - Super noeud
 - paramètres, 26, 34
 - Super noeuds
 - définition de propriétés dans, 320
 - génération de scripts, 320
 - paramètres, 320
 - propriétés, 320
 - scripts, 9, 15–16
 - syntaxe avec caret
 - références de variable, 21, 25
 - système
 - arguments de ligne de commande, 75
-
- Tableaux de bord
 - création et utilisation de scripts, 65, 68
-
- Variables, 25, 34
 - génération de scripts, 20, 23
 - références de noeud, 21
 - variables locales, 25, 34
 - variables spéciales, 23