

IBM SPSS Modeler 15 脚本编写
和自动化指南



注意：使用本信息及其支持的产品之前，请阅读注意事项第 页码下的一般信息。

此版本适用于 IBM SPSS Modeler 15 及所有后续发布和修订，除非在新版本中另有说明。

Adobe 产品屏幕截图重印已获得 Adobe Systems Incorporated 的许可。

Microsoft 产品屏幕截图重印已获得 Microsoft Corporation 的许可。

受许可保护材料 - IBM 所有

Copyright IBM Corporation 1994, 2012.

美国政府用户受限权利 - 使用、复制或披露受与 IBM Corp. 签订的 GSA ADP Schedule Contract 的限制。

前言

IBM® SPSS® Modeler 是 IBM Corp. 企业级数据挖掘工作平台。SPSS Modeler通过深入的数据分析帮助组织改进与客户和市民的关系。组织通过借助源自 SPSS Modeler 的洞察力可以留住优质客户，识别交叉销售机遇，吸引新客户，检测欺诈，降低风险，促进政府服务交付。

SPSS Modeler’ 的可视化界面让用户可以应用他们自己的业务专长，这将生成更强有力的预测模型，缩减实现解决方案所需的时间。SPSS Modeler 提供了多种建模技术，例如预测、分类、细分和关联检测算法。模型创建成功后，通过 IBM® SPSS® Modeler Solution Publisher，在广泛的企业内交付给决策者，或通过数据库交付。

关于 IBM Business Analytics

IBM Business Analytics 软件为决策者提供可信赖的完整、一致和准确信息，以帮助其提升业务绩效。这一涵盖 [商务智能](#)、[预测分析](#)、[财务绩效与战略管理](#)以及[分析应用程序](#)的全面组合可提供有关当前业务表现的清晰、立即和切实可行的深入见解，并能够有效预测未来结果。其中整合了丰富的行业解决方案、经过验证的做法与专业服务，以帮助各种规模的组织提升生产效率、自动化决策并取得卓越成果。

作为该软件组合的一部分，IBM SPSS Predictive Analytics 软件能够帮助各类组织有效地预测未来事件，并针对所得到的深入见解提前采取行动，以取得更优秀的业务成果。全球企业、政府和学院客户依赖 IBM SPSS 技术作为吸引、留住和增加客户数量的竞争优势，并降低欺诈和转移风险。通过将 IBM SPSS 软件融入其日常运营中，这些组织将成为“预测型”企业，即能够指引并自动化决策，以实现业务目标和取得可衡量的竞争优势。有关详细信息，或联系我们的代表，请访问 <http://www.ibm.com/spss>。

技术支持

我们提供有技术支持服务以维护客户。客户可就 IBM Corp. 产品使用或某一受支持硬件环境的安装帮助寻求技术支持。要获得技术支持，请访问 IBM Corp. 网站 <http://www.ibm.com/support>。在请求帮助时，请做好准备，以便识别您自己、您的组织以及您的支持协议。

内容

1 关于 IBM SPSS Modeler 1

IBM SPSS Modeler 产品	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	2
IBM SPSS Modeler Server 适配器用于 IBM SPSS Collaboration and Deployment Services	2
IBM SPSS Modeler 版本	3
IBM SPSS Modeler 文档	3
SPSS Modeler Professional 文档	4
SPSS Modeler Premium 文档	4
应用程序示例	5
Demos 文件夹	6

部分 I: 脚本编写和脚本编写语言

2 脚本编写概述 8

脚本类型	8
流脚本	8
流脚本示例: 训练神经网络	9
独立脚本	10
独立脚本示例: 保存和载入模型	11
独立脚本示例: 生成特征选择模型	12
超节点脚本	13
超节点脚本示例	14
执行和中断脚本	14
查找和替换	15

3 脚本编写语言 18

脚本编写语言概述	18
脚本编写语法	18
引用节点	19

检索对象	20
设置当前对象	20
打开流和其他对象	21
使用多个流	22
局部脚本变量	22
流、会话和超节点参数	23
控制脚本执行	24
脚本中的操作符	25
CLEM 脚本中的表达式	25
插入注释和连接符	26
文字文本块	26

4 脚本编写命令 28

一般脚本编写命令	28
execute_all	28
execute_script	28
exit	28
for...endfor	29
if...then...else...	30
set 命令	30
var 命令	33
节点对象	33
create NODE	33
connect NODE	34
delete NODE	35
disable NODE	35
disconnect NODE	35
duplicate NODE	35
enable NODE	35
execute NODE	36
export NODE as FILE	36
flush NODE	37
get node NODE	37
load node FILENAME	37
position NODE	37
rename NODE as NEWNAME	38
retrieve node REPOSITORY_PATH	38
save node NODE as FILENAME	38
store node NODE as REPOSITORY_PATH	39

模型对象	39
模型块名称	39
避免重复的模型名称	41
delete model MODEL	41
export model MODEL as FILE	42
insert model MODEL	43
load model FILENAME	43
retrieve model REPOSITORY_PATH	43
save model MODEL as FILENAME	44
store model MODEL as REPOSITORY_PATH	44
流对象	44
create stream DEFAULT_FILENAME	44
close STREAM	44
clear stream	45
get stream STREAM	45
load stream FILENAME	45
open stream FILENAME	46
retrieve stream REPOSITORY_PATH	46
save STREAM as FILENAME	46
store stream as REPOSITORY_PATH	47
with stream STREAM	47
工程对象	48
execute_project	48
load project FILENAME	48
retrieve project REPOSITORY_PATH	48
save project as FILENAME	48
store project as REPOSITORY_PATH	48
状态对象	49
load state FILENAME	49
结果对象	49
value RESULT	49
文件对象	49
close FILE	49
open FILE	50
write FILE	50
输出对象	50
输出类型名称	50
delete output OUTPUT	51
export output OUTPUT	51
get output OUTPUT	51
load output FILENAME	52
retrieve output REPOSITORY_PATH	52

save output OUTPUT as FILENAME	52
store output OUTPUT as REPOSITORY_PATH.	52

5 脚本编写技巧 53

修改流执行	53
对节点执行循环	53
访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象	54
生成加密密码	55
脚本检查	56
从命令行编写脚本.	56
与早期版本的兼容性	56

6 脚本编写示例 58

类型节点报告	58
流报告	61

7 命令行参数 63

调用软件	63
命令行参数的使用.	63
组合多个参数	64
服务器连接参数	64
IBM SPSS Collaboration and Deployment Services Repository 连接参数	65
系统参数.	66
形参的实参值	67

8 CLEM 语言参考 69

CLEM参考概述	69
CLEM数据类型	69
整数	70
实数	70
字符	70
字符串	70

列表	71
字段	71
日期	71
时间	72
CLEM运算符	73
函数参考	74
函数说明中的约定	75
信息函数	76
转换函数	76
比较函数	77
逻辑函数	79
数值函数	79
三角函数	80
概率函数	81
位元整数运算	81
随机函数	82
字符串函数	82
SoundEx 函数	86
日期和时间函数	87
序列函数	90
全局函数	94
处理空值和 Null 值的函数	94
特殊字段	95

部分 II：属性参考信息

9 属性参考信息	98
属性参考信息概述	98
属性语法	98
节点和流属性示例	100
节点属性概述	100
通用节点属性	100

10 流属性	102
11 工程属性	105
12 源节点属性	106
源节点通用属性	106
cognosimportnode 属性	108
databasenode 属性	110
datacollectionimportnode 属性	111
excelimportnode 属性	113
evimportnode 属性	114
fixedfilenode 属性	115
sasimportnode 属性	117
statisticsimportnode 属性	117
userinputnode 属性	118
variablefilenode 属性	119
xmlimportnode 属性	121
13 记录操作节点属性	123
appendnode 属性	123
aggregatenode 属性	123
balancenode 属性	124
distinctnode 属性	125
mergenode 属性	125
rfmaggregatenode 属性	127
samplenode 属性	128
selectnode 属性	130
sortnode 属性	130
14 字段操作节点属性	131
anonymizenode 属性	131
autodatapreprenode 属性	132

binningnode 属性	135
derivenode 属性	137
ensemblenode 属性	138
fillernode 属性	139
filternode 属性	140
historynode 属性	141
partitionnode 属性	142
reclassifynode 属性	143
reordernode 属性	144
restructurenode 属性	144
rfanalysisnode 属性	145
settoflagnode 属性	146
statisticstransformnode 属性	147
timeintervalnode 属性	147
transposenode 属性	151
typenode 属性	152

15 图形节点属性 157

图形节点通用属性	157
collectionnode 属性	158
distributionnode 属性	159
evaluationnode 属性	159
graphboardnode 属性	161
histogramnode 属性	163
multiplotnode 属性	164
plotnode 属性	165
timeplotnode 属性	167
webnode 属性	168

16 建模节点属性 171

通用建模节点属性	171
anomalydetectionnode 属性	171
apriorinode 属性	172
autoclassifiernode 属性	174
设置算法属性	175

autoclusternode 属性	176
autonumericnode 属性	178
bayesnetnode 属性	179
c50node 属性	180
carmanode 属性.	181
cartnode 属性	183
chaidnode 属性.	185
coxregnode 属性	187
decisionlistnode 属性.	189
discriminantnode 属性.	190
factornode 属性	191
featureselectionnode 属性	193
genlinnode 属性	194
GLMM 节点属性	197
kmeansnode 属性	200
knnnode 属性	201
kohonenode 属性.	202
linearnode 属性	204
logregnode 属性	205
neuralnetnode 属性	209
neuralnetworknode 属性	211
questnode 属性.	213
regressionnode 属性.	215
sequencenode 属性	217
slrmnode 属性	218
statisticsmodelnode 属性	219
svmnode 属性	219
timeseriesnode 属性.	220
twostepnode 属性	222

17 模型块节点属性 224

applyanomalydetectionnode 属性	224
applyapriorinode 属性.	224
applyautoclassifiernode 属性	225
applyautoclusternode 属性	225
applyautonumericnode 属性	225
applybayesnetnode 属性	226

applyc50node 属性	226
applycarmanode 属性.	226
applycartnode 属性	226
applychaidnode 属性.	227
applycoxregnode 属性	227
applydecisionlistnode 属性.	228
applydiscriminantnode 属性.	228
applyfactornode 属性	228
applyfeatureselectionnode 属性	228
applygeneralizedlinearnode 属性.	229
applykmeansnode 属性	229
applyknnnode 属性	229
applykohonennode 属性.	229
applylinearnode 属性	229
applylogregnode 属性	230
applyneuralnetnode 属性	230
applyneuralnetworknode 属性	230
applyquestnode 属性.	231
asapplyregressionnode 属性.	231
applyselflearningnode 属性.	231
applysequencenode 属性	231
applysvmnode 属性	232
applytimeseriesnode 属性	232
applytwostepnode 属性	232

18 数据库建模节点属性 233

Microsoft 建模的节点属性	233
Microsoft 建模节点属性	233
Microsoft 模型块属性	236
Oracle 建模的节点属性.	238
Oracle 建模节点属性	238
Oracle 模型块属性	243
IBM DB2 建模的节点属性	244
IBM DB2 建模节点属性	244
IBM DB2 模型块属性	249
IBM Netezza Analytics 建模节点属性	250
Netezza 建模节点属性.	250
Netezza 模型块属性	256

19 输出节点属性 257

analysisnode 属性	257
dataauditnode 属性	258
matrixnode 属性	259
meansnode 属性	261
reportnode 属性	262
setglobalsnode 属性	263
statisticsnode 属性	264
statisticsoutputnode 属性	265
tablenode 属性	265
transformnode 属性	268

20 导出节点属性 269

通用导出节点属性	269
cognosexportnode 属性	269
databaseexportnode 属性	270
datacollectionexportnode 属性	274
excelexportnode 属性	274
outputfilenode 属性	275
sasexportnode 属性	276
statisticsexportnode 属性	277
xmlexportnode 属性	277

21 IBM SPSS Statistics 节点属性 278

statisticsimportnode 属性	278
statisticstransformnode 属性	278
statisticsmodelnode 属性	279
statisticsoutputnode 属性	279
statisticsexportnode 属性	280

22 超节点属性 281

附录

A 注意事项 283

索引 285

关于 IBM SPSS Modeler

IBM® SPSS® Modeler 是一组数据挖掘工具，通过这些工具可以采用商业技术快速建立预测性模型，并将其应用于商业活动，从而改进决策过程。SPSS Modeler 参照行业标准 CRISP-DM 模型设计而成，可支持从数据到更优商业成果的整个数据挖掘过程。

SPSS Modeler 提供了各种借助机器学习、人工智能和统计学的建模方法。通过建模选项板中的方法，您可以根据数据生成新的信息以及开发预测模型。每种方法各有所长，同时适用于解决特定类型的问题。

SPSS Modeler 可以作为独立产品购买，或作为客户端与 SPSS Modeler Server 一起使用。同时提供了大量其他选项，以下各节将对这些选项进行概述。有关详细信息，请参阅 <http://www.ibm.com/software/analytics/spss/products/modeler/>。

IBM SPSS Modeler 产品

IBM® SPSS® Modeler 系列产品及其相关软件包括如下成员：

- 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 适配器用于 IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler 是在个人电脑上安装并运行的完整功能版本产品。您可以在本地模式下将 SPSS Modeler 作为独立产品来运行；也可以在分布式模式下与 IBM® SPSS® Modeler Server 协同使用，从而提高了对大数据集的处理速度。

使用 SPSS Modeler，您可以快速直观地建构精确的预测模型，无需编程。使用其独特的可视界面，您可以轻松地将数据挖掘过程可视化。通过内嵌在该产品中的高级分析支持，您可以发现之前在您的数据中隐藏的模式与趋势。您可以建构输出模型，并理解其影响因子，让您可以更好地利用业务机会、降低风险。

SPSS Modeler 提供两个版本：SPSS Modeler Professional 和 SPSS Modeler Premium。有关详细信息，请参阅 [IBM SPSS Modeler 版本中的 IBM SPSS Modeler 15 用户指南](#)。

IBM SPSS Modeler Server

SPSS Modeler 使用客户端/服务器体系结构将资源集约型操作的请求分发给功能强大的服务器软件，因而使大数据集的传输速度大大加快。

SPSS Modeler Server 是一项独立许可的产品，可以协同一份或多份 IBM® SPSS® Modeler 安装，以分布式分析模式在服务器主机上持续运行。因此，SPSS Modeler Server 在处理大型数据集时具有卓越的性能，因为内存密集型的操作可在服务器完成而无需将数据下载到客户端计算机。IBM® SPSS® Modeler Server 还提供了对 SQL 优化的支持与数据库内建模能力，有助于进一步提高性能与自动化程度。

IBM SPSS Modeler Administration Console

Modeler Administration Console 是一个图形化的应用程序，可管理 SPSS Modeler Server 的多项配置选项，配置也可通过编辑一个选项文件来进行。该应用程序提供了一个控制台用户界面，用以监控和配置所安装的 SPSS Modeler Server，SPSS Modeler Server 的现用户可以免费使用该应用程序。应用程序只能安装在 Windows 计算机上；但是它可以管理安装在任何受支持平台上的服务器。

IBM SPSS Modeler Batch

虽然数据挖掘通常是交互式的过程，但是也可以通过命令行来运行 SPSS Modeler 而无需图形用户界面。例如，您可能有些任务需长期或重复性地运行而无用户干预。SPSS Modeler Batch 是该产品一个特殊版本，无需通过常规的用户界面即可完整地实现 SPSS Modeler 的分析功能。使用 SPSS Modeler Batch 需要具备 SPSS Modeler Server 许可。

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher 是一个工具，让您可以创建 SPSS Modeler 流的打包版本，由外部运行时引擎运行，或嵌入一个外部应用程序。以此方式，您可以发布与部署完整的 SPSS Modeler 流，在未安装 SPSS Modeler 的环境下也能使用。SPSS Modeler Solution Publisher 是作为 IBM SPSS Collaboration and Deployment Services - Scoring 服务的一部分来发行的，需另行购买许可。获得许可后，您会收到 SPSS Modeler Solution Publisher Runtime，让您可以执行已发布的流。

IBM SPSS Modeler Server 适配器用于 IBM SPSS Collaboration and Deployment Services

有若干种 IBM® SPSS® Collaboration and Deployment Services 适配器可以让您通过 SPSS Modeler 和 SPSS Modeler Server 与 IBM SPSS Collaboration and Deployment Services 存储库交互。以此方式，部署在存储库中的 SPSS Modeler 流即可实现多用户共享，或通过瘦客户端应用程序 IBM SPSS Modeler Advantage 访问。适配器须安装在存储库的主机系统中。

IBM SPSS Modeler 版本

SPSS Modeler 提供下列版本。

SPSS Modeler Professional

SPSS Modeler Professional 提供您在处理大多数类型的结构化数据（例如 CRM 系统中跟踪的行为或交互活动、人口统计、购买行为与销售数据）时所需的所有工具。

SPSS Modeler Premium

SPSS Modeler Premium 是一项单独许可的产品，它扩展了 SPSS Modeler Professional 的功能，使其可以处理如实体分析或社会网络专门数据，以及非结构化的文本数据。SPSS Modeler Premium 包含下列组件。

IBM® SPSS® Modeler Entity Analytics 在 IBM® SPSS® Modeler 预测分析的基础上添加了全新的维度。预测分析会尝试根据过去数据预测未来行为，而实体分析侧重于通过解析记录自身的身份冲突，提高当前数据的连贯性和一致性。身份可以指个人、组织、对象或可能不确定的任何其他实体的身份。身份解析在许多领域中都非常重要，包括客户关系管理、检测、反洗钱以及国家与国际安全。

IBM SPSS Modeler Social Network Analysis 将关于关系的信息转换为字段，这些字段可描述个人和组社交行为的特征。使用介绍社交网络之下关系的数据，IBM® SPSS® Modeler Social Network Analysis 可识别影响网络中他人行为的社交领导。此外，可确定受其他网络参与者影响最大的人员。通过结合这些结果和其他测量，您可创建个人的综合配置文件，作为预测模型的基础。包括此社交信息的模型比不包括的模型执行效果更好。

Text Analytics for IBM® SPSS® Modeler 采用了先进语言技术和 Natural Language Processing (NLP)，以快速处理大量无结构文本数据，抽取和组织关键概念，以及将这些概念分为各种类别。抽取的概念和类别可以和现有结构化数据中进行组合（例如人口统计学），并且可用于借助 SPSS Modeler 的一整套数据挖掘工具来进行建模，以此实现更好更集中的决策。

IBM SPSS Modeler 文档

可以从 SPSS Modeler 的帮助菜单中获取在线帮助格式的文档。此文档包括 SPSS Modeler、SPSS Modeler Server 和 SPSS Modeler Solution Publisher 的文档以及《应用程序指南》和其他支持材料。

每个产品的完整文档（包括安装说明）也在每个产品 DVD 的 \Documentation 文件夹下以 PDF 格式提供。安装文档也可从以下网页中下载：
<http://www-01.ibm.com/support/docview.wss?uid=swg27023172>。

两种格式的文档均可从 SPSS Modeler 信息中心获取，其网址如下：
<http://publib.boulder.ibm.com/infocenter/spssmodl/v15r0m0/>。

SPSS Modeler Professional 文档

SPSS Modeler Professional 的文档套件（不含安装说明）如下：

- **IBM SPSS Modeler 用户指南。** 使用 SPSS Modeler 的一般使用介绍，包括如何构建数据流、处理缺失值、生成 CLEM 表达式、处理项目和报告以及将用于部署的流打包为 IBM SPSS Collaboration and Deployment Services、Predictive Application 或 IBM SPSS Modeler Advantage。
- **IBM SPSS Modeler 源、处理和输出节点。** 介绍用于以不同的格式读取、处理和输出数据的所有节点。实际上这表示所有节点而非建模节点。
- **IBM SPSS Modeler 建模节点。** 有关用于创建数据挖掘模型的所有节点的描述。IBM® SPSS® Modeler 可提供各种借助机器学习、人工智能和统计学的建模方法。 [有关详细信息，请参阅第 3 章中的建模节点概述中的 IBM SPSS Modeler 15 建模节点。](#)
- **IBM SPSS Modeler 算法指南。** 介绍 SPSS Modeler 中所用建模方法的数学基础。此指南仅提供 PDF 版。
- **IBM SPSS Modeler 应用程序指南。** 本指南中的示例旨在为具体的建模方法和技术提供具有针对性的简介。还可以在“帮助”菜单中查阅本指南的在线版本。 [有关详细信息，请参阅应用程序示例中的 IBM SPSS Modeler 15 用户指南。](#)
- **IBM SPSS Modeler 脚本编写与自动化。** 通过编写脚本实现系统自动化的相关信息，包括用于操作节点和流的属性信息。
- **IBM SPSS Modeler 部署指南。** 有关在 IBM® SPSS® Collaboration and Deployment Services Deployment Manager 中以处理作业的步骤形式运行 SPSS Modeler 流和方案的信息。
- **IBM SPSS Modeler CLEF 开发人员指南** CLEF 提供了将第三方程序（例如，数据处理例程或建模算法）作为节点集成到 SPSS Modeler 的功能。
- **IBM SPSS Modeler 数据库内数据挖掘指南。** 有关如何利用数据库的功能通过第三方算法来改进性能并增强分析功能的信息。
- **IBM SPSS Modeler Server 管理和性能指南。** 有关如何配置和管理 IBM® SPSS® Modeler Server 的信息。
- **IBM SPSS Modeler Administration Console 用户指南。** 有关安装和使用控制台用户界面以监视和配置 SPSS Modeler Server 的信息。控制台实现为 Deployment Manager 应用程序的插件。
- **IBM SPSS Modeler Solution Publisher 指南。** SPSS Modeler Solution Publisher 是一个附加式组件，通过它组织可发布在标准 SPSS Modeler 环境之外使用的流。
- **IBM SPSS Modeler CRISP-DM 指南。** 借助 CRISP-DM 方法进行 SPSS Modeler 数据挖掘的分步指南。
- **IBM SPSS Modeler Batch 用户指南。** 在批处理模式下使用 IBM SPSS Modeler 的完整指南，包括批处理模式的执行与命令行参数的详细信息。此指南仅提供 PDF 版。

SPSS Modeler Premium 文档

SPSS Modeler Premium 的文档套件（不含安装说明）如下：

- **IBM SPSS Modeler Entity Analytics 用户指南。** 关于通过 SPSS Modeler 使用实体分析的信息，涵盖存储库的安装与配置、实体分析节点以及管理任务。

- **IBM SPSS Modeler Social Network Analysis 用户指南。** 通过 SPSS Modeler 进行社会网络分析的指南，包括群组分析与传播分析。
- **Text Analytics for SPSS Modeler 用户指南。** 关于通过 SPSS Modeler 使用文本分析的信息，涵盖文本发掘节点、交互式工作台、模板及其他资源。
- **Text Analytics for IBM SPSS Modeler Administration Console 用户指南。** 有关安装和使用控制台用户界面监视和配置 IBM® SPSS® Modeler Server 用于 Text Analytics for SPSS Modeler 的信息。控制台实现为 Deployment Manager 应用程序的插件。

应用程序示例

SPSS Modeler 中的数据挖掘工具可以帮助解决很多业务和组织问题，应用程序示例将提供有关特定建模方法和技术的简明的针对性说明。此处使用的数据集比某些数据挖掘器管理的大量数据存储要小得多，但涉及的概念和方法应可扩展到实际的应用程序。

可以通过在 SPSS Modeler 中的“帮助”菜单中单击[应用程序示例](#)来访问示例。数据文件和样本流安装在产品安装目录下的 Demos 文件夹中。[有关详细信息，请参阅Demos 文件夹中的IBM SPSS Modeler 15 用户指南。](#)

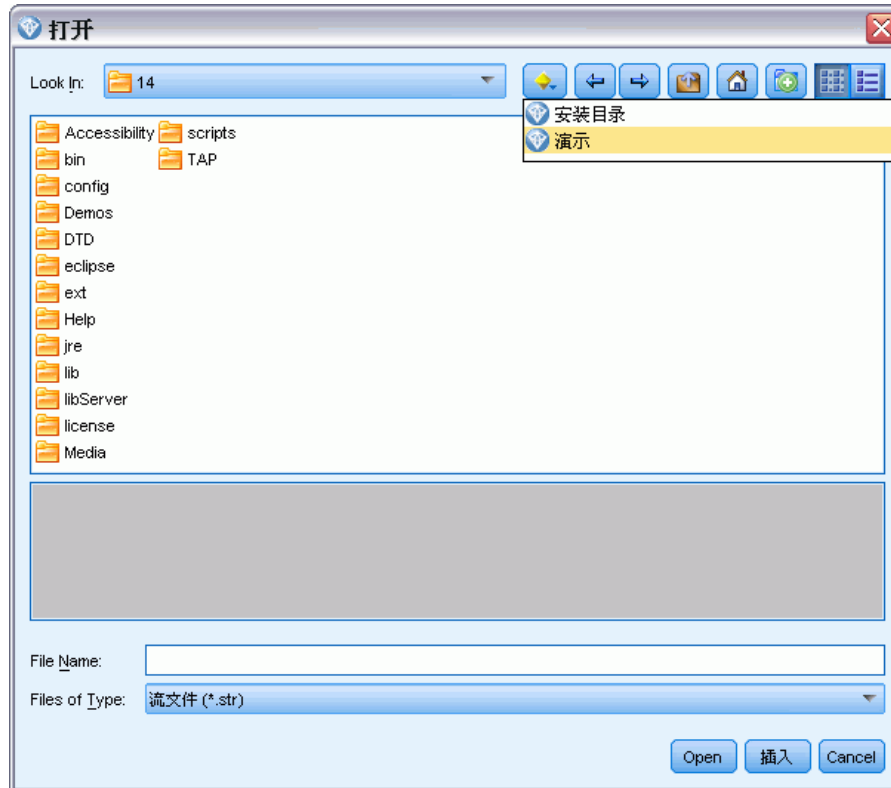
数据库建模示例。 请参阅 IBM SPSS Modeler 数据库内挖掘指南 中的示例。

编写示例脚本。 请参阅 IBM SPSS Modeler 脚本编写和自动化指南 中的示例。

Demos 文件夹

与应用程序示例一起使用的数据文件和样本流安装在产品安装目录下的 Demos 文件夹中。可从 Windows 的“开始”菜单中 IBM SPSS Modeler 15 程序组访问该文件夹，也可以在“文件打开”对话框中最近目录的列表中单击 Demos。

图片 1-1
在最近使用的目录列表中选择 Demos 文件夹



部分 I: 脚本编写和脚本编写语言

脚本编写概述

IBM® SPSS® Modeler 中的脚本编写是用于在用户界面上实现过程自动化的强大工具。您使用鼠标或键盘进行的操作, 借助脚本同样可以完成, 而且使用脚本可以自动化那些手动执行将造成大量重复操作且高耗时的任务。

脚本的作用包括:

- 限制在流中执行节点的特定顺序。
- 设置节点属性并使用 CLEM (表达式操作控制语言) 的子集来执行派生。
- 指定通常包含用户交互的操作的自动执行顺序, 例如您可以构建一个模型, 然后对其进行测试。
- 设置需要实际用户交互的复杂过程, 例如需要重复模型生成和测试的交叉验证步骤。
- 设置流操纵过程—例如, 您可以提取一个模型训练流, 运行它, 然后自动生成相应的模型测试流。

这一的 章节提供流级脚本、独立脚本以及 SPSS Modeler 用户界面超节点内脚本的高级说明和示例。有关脚本编写语言、语法和命令的更多信息, 请参阅章后的章节。

注意: 您无法导入和运行在 SPSS Modeler 中的 IBM® SPSS® Statistics 中创建的脚本。

脚本类型

IBM® SPSS® Modeler 采用了三种类型的脚本:

- **流脚本** 存储为流属性然后和指定流一起保存和载入。例如, 可以编写自动化训练和应用模型块流程的流脚本。还可以指定何时执行特定流, 脚本应代替流工作区内容运行。
- **独立脚本** 和保存在外部文本文件中的所有特定流关联。例如, 可以使用独立脚本同时操作多个流。
- **超节点脚本** 存储为超节点流属性。超节点只在终端超节点中可用。可以使用超节点脚本控制超节点内容的执行序列。对于非终端 (源或过程) 超节点, 可以为超节点定义属性或定义这种超节点直接在流脚本中包含的节点。

流脚本

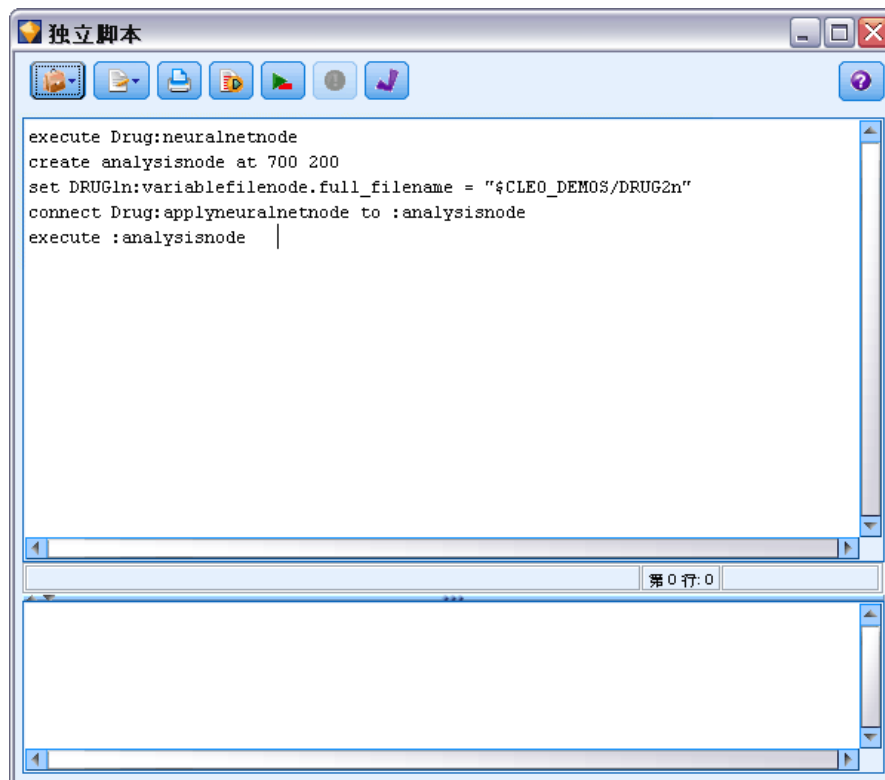
脚本可用于自定义特定流中的操作并与该流一起保存。流脚本可用于指定某个流中终端节点的特定执行顺序。可以使用“流脚本”对话框来编辑与当前流一起保存的脚本。

从“流属性”对话框访问流“脚本”选项卡:

- ▶ 从“工具”菜单中, 选择:
流属性 > 脚本...

- ▶ 单击脚本选项卡，处理当前流的脚本。

图片 2-1
流脚本对话框



该对话框顶部的工具栏图标允许您执行以下操作：

- 将先前存在的独立脚本内容导入到窗口。
- 将脚本保存为文本文件。
- 打印脚本。
- 追加默认脚本。
- 执行整个当前流。
- 执行某个脚本中的选定行。
- 检查脚本的语法，如果发现任何错误，就将其显示在对话框的下部面板中复查。

此外，也可以指定当执行流时是否应运行此脚本。每当按照脚本的执行顺序执行流时，您可以选择运行该脚本来运行脚本。此设置为快速构建模型提供流一级的自动化。但是，默认设置为在执行流的过程中忽略此脚本。即使选择选项忽略此脚本，也可以直接从此对话框运行脚本。

流脚本示例：训练神经网络

当执行时，流可用于训练神经网络模型。通常，要测试模型，您可以运行建模节点以将模型添加到流，并做相应连接，然后执行分析节点。

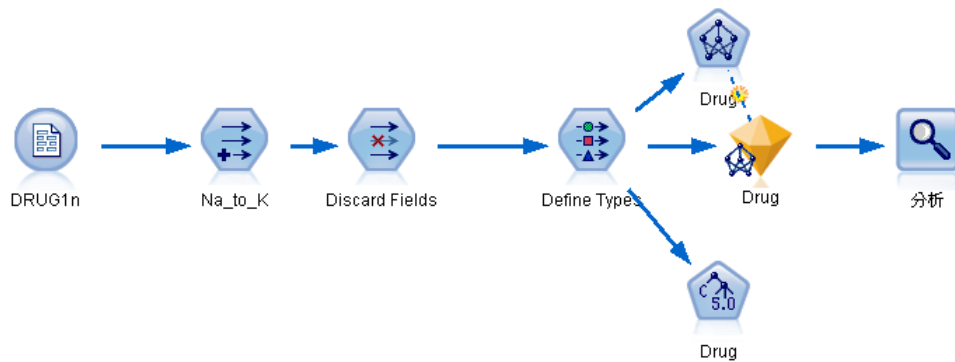
借助 IBM® SPSS® Modeler 脚本，您可以在创建模型块之后，实现模型块测试过程的自动化。例如，以下流脚本将测试演示流 druglearn.str（在 SPSS Modeler 安装下的 /Demos/streams/ 文件夹中），并可从“流属性”对话框（工具 > 流属性 > 脚本）中运行。

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

以下带着重号的句子说明此脚本示例中的每一行。

- 第一行执行已在演示流中发现的名为 **Drug** 的神经网络节点，以便创建模型块，并将其放置在已连接到流中类型节点的流工作区中。
- 在第二行中，脚本创建了一个分析节点并将其放入 700 x 200 的工作区位置。
- 在第三行中，将流中使用的原始数据源切换到名为 **DRUG2n** 的测试数据集。
- 在第四行中，神经网络模型块被连接到分析节点。请注意，因为流中不存在其他相似的节点，所以没有用名称指示神经网络模型块和分析节点。
- 最后，执行分析节点以生成分析报告。

图片 2-2
最终生成流



请注意，该脚本用于处理现有流，因为它假定神经网络节点 Drug 已经存在。但还可以使用脚本来构建并从开始位置运行流（从空工作区开始）。要了解有关常规脚本编写语言的更多信息，请参阅脚本编写语言概述第 18 页码。要了解有关具体脚本编写命令的更多信息，请参阅脚本编写命令第 28 页码。

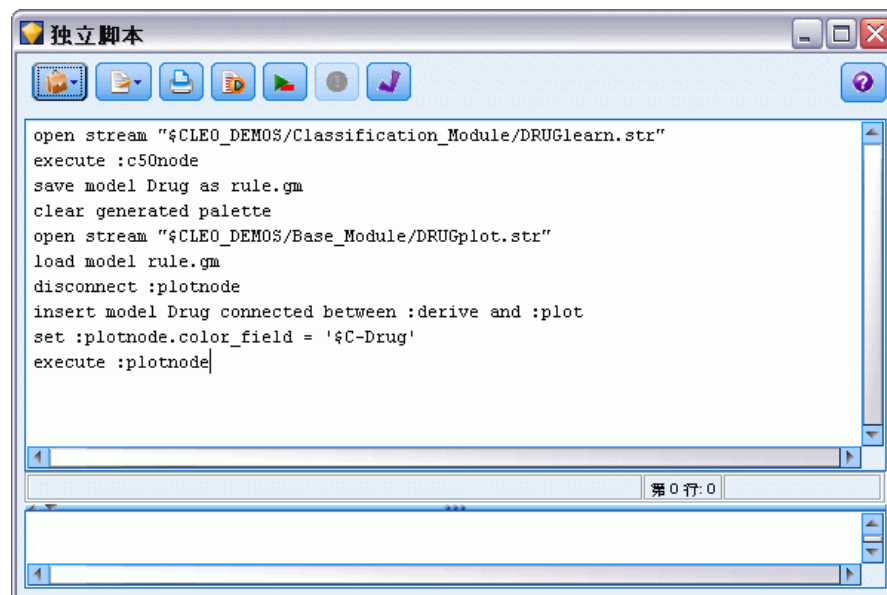
独立脚本

“独立脚本”对话框用于创建或编辑保存为文本文件的脚本。它显示了文件名称，提供了用于载入、保存、导入和执行脚本的实用程序。

要访问“独立脚本”对话框，请执行以下操作：

- ▶ 在主菜单中，选择：
工具 > 独立脚本

图片 2-3
独立脚本对话框



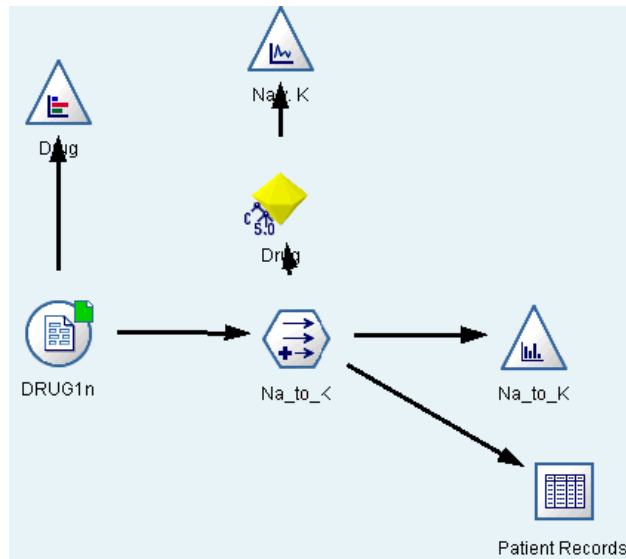
对流脚本可用的工具栏和脚本语法检查选项对独立脚本同样适用。 [有关详细信息，请参阅第 8 页码流脚本。](#)

独立脚本示例：保存和载入模型

独立脚本可用于流操纵。假设有两个流，第一个流创建模型，第二个流则通过现有数据字段，采用图示的方式对来自第一个流的规则集进行探索。该方案的独立脚本可能具有如下形式：

```
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
```

图片 2-4
最终生成流



注意：要了解有关常规脚本编写语言的更多信息，请参阅 脚本编写语言概述第 18 页码。要了解有关具体脚本编写命令的更多信息，请参阅脚本编写命令第 28 页码。

独立脚本示例：生成特征选择模型

首先打开一个空工作区，在此示例中将构建一个流，该流生成一个特征选择模型，应用此模型并创建一个表，该表包含有对于指定目标而言重要性最高的 15 个字段。

```
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
```

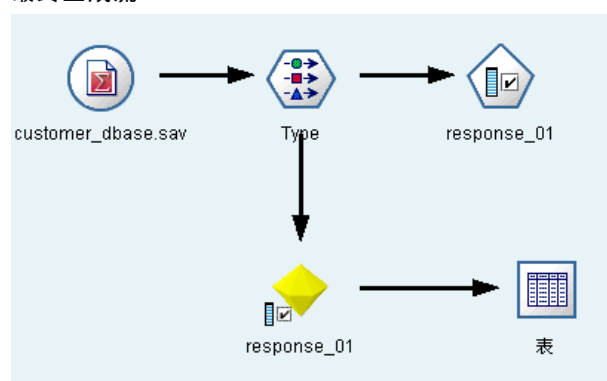
```

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

```

此脚本创建了一个用以读入数据的源节点，使用“类型”节点将字段 response_01 的角色（方向）设置为Target，然后创建并执行“特征选择”节点。此脚本还连接各节点并将其置于流工作区中，以此生成可读布局。然后结果模型块与表节点相连接，“表”节点列出了属性 selection_mode 和 top_n 所确定的 15 个最重要的字段。[有关详细信息，请参阅第 193 页码第 16 章中的featureselectionnode 属性。](#)

图片 2-5
最终生成流



超节点脚本

通过使用 IBM® SPSS® Modeler 脚本语言，可以创建和保存所有终端超节点中的脚本。这些脚本只在终端超节点中可用，并且常在创建模板流或用于强制超节点内容以特定顺序执行时使用。使用超节点脚本，您也可以在此流中运行多个脚本。

例如，假设需要指定一个复杂流的执行顺序，并且超节点包含若干个包括设置全局量节点的节点，而执行设置全局量节点又需要在派生用于散点图节点的新字段之前进行。这种情况下，可以创建一个首先执行设置全局量节点的超节点脚本。由设置全局量节点计算出的值，例如平均差或标准差，可在散点图节点的执行过程中使用。

在超节点脚本中也可以指定节点属性，操作方法与在其他脚本中的进行的操作一样。或者，为所有超节点或直接来自流脚本的超节点的封装节点更改和定义属性。[有关详细信息，请参阅第 281 页码第 22 章中的超节点属性。](#)此方法适用于源和过程超节点以及终端超节点。

注意：因为只有终端超节点能够执行自身脚本，所以“超节点”对话框的“脚本”选项卡只在用于终端超节点时可用。

从主工作区打开“超节点脚本”对话框：

- ▶ 从流工作区选择终端超节点，然后从“超节点”菜单选择：
超节点脚本...

从放大超节点工作区打开“超节点脚本”对话框：

- ▶ 右键单击超节点工作区，然后从上下文菜单中选择：
超节点脚本...

有关详细信息，请参阅第 9 章中的超节点和脚本编写中的 IBM SPSS Modeler 15 源、过程和输出节点。

超节点脚本示例

以下超节点脚本声明执行超节点中终端节点的顺序。此顺序可确保首先执行设置全局量节点，以便随后执行其他节点时可使用由此节点计算的值。

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

执行和中断脚本

存在多种执行脚本的方式。例如，在流脚本或独立脚本对话框中，“运行此脚本”按钮执行完整的脚本：

图片 2-6
“运行此脚本”按钮



“运行选定行”按钮执行在脚本中选定的单行或相邻行块：

图片 2-7
“运行选定行”按钮



可以使用以下方式执行脚本：

- 在流脚本或独立脚本对话框中，单击“运行此脚本”或“运行选定行”按钮。
- 在运行此脚本设置为默认执行方式的情况下运行流。
- 启动后以交互模式使用 `-execute` 标志。有关详细信息，请参阅第 63 页码第 7 章中的命令行参数的使用。

注意：如果在“超节点脚本”对话框中选择运行此脚本，则将在执行超节点时执行超节点脚本。

中断脚本执行

“流脚本”对话框中的红色“停止”按钮将在脚本执行过程中被激活。使用此按钮可以放弃脚本和任何当前流的执行。

查找和替换

可在编辑脚本或表达式文本的位置（包括脚本编辑器和 CLEM 表达式构建器）或定义“报告”节点中的模板时使用“查找/替换”对话框。在上述任一区域编辑文本时，按 Ctrl+F，可访问此对话框，并确保光标的焦点位于文本区域中。例如，处理填充节点时，可以通过“设置”选项卡的任一文本区域或表达式构建器中的文本字段访问此对话框。

图片 2-8
“查找/替换”对话框



- ▶ 使用文本区域中的光标，按 Ctrl+F 可访问“查找/替换”对话框。
- ▶ 输入要搜索的文本，或从最近搜索项下拉列表中选择。
- ▶ 输入替换文本（如果有的话）。
- ▶ 单击查找下一个开始搜索。
- ▶ 单击替换替换当前选定的内容，或单击全部替换更新所有项或选定的实例。
- ▶ 每次操作完成后，此对话框将关闭。从任一文本区域中按 F3 键，可重复上一次查找操作，或按 Ctrl+F，可再次访问该对话框。

搜索选项

区分大小写。指定查找操作是否区分大小写；例如 myvar 是否匹配 myVar。无论怎样设置，替换文本始终完全按照输入插入。

仅限于整个单词。指定查找操作是否匹配单词中内嵌的文本。如果选中，spider 的搜索结果将不会包括 spiderman 或 spider-man。

正则表达式。指定是否使用正则表达式语法（请参阅下一节）。如果选中，仅限于整个单词选项将禁用并且会忽略其值。

仅限于选择文本。使用全部替换选项时，控制搜索范围。

正则表达式语法

使用正则表达式，您可以搜索特殊字符（如选项卡或换行字符）、字符的类或范围（如 a 到 d）、任何数字或非数字以及边界（如行首或行尾）。支持的表达式类型如下。

字符匹配

字符	匹配项
x	字符 x
\\	反斜杠字符
\On	含八进制值的字符 On (0 <= n <= 7)
\Onn	含八进制值的字符 Onn (0 <= n <= 7)
\Omnn	含八进制值的字符 Omnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	含十六进制值的字符 Oxhh
\uhhhh	含十六进制值的字符 Oxhhhh
\t	制表符 ('\u0009')
\n	新建行 (换行) 符 ('\u000A')
\r	回车符 ('\u000D')
\f	换页符 ('\u000C')
\a	警报 (铃声) 字符 ('\u0007')
\e	转义符 ('\u001B')
\cx	x 对应的控制字符

匹配字符类

字符类	匹配项
[abc]	a、b、或 c (简单类)
[^abc]	除 a、b、或 c 之外的所有字符 (相减)
[a-zA-Z]	a 到 z 或 A 到 Z, 包含 (范围)
[a-d[m-p]]	a 到 d 或 m 到 p (合并)。也可指定为 [a-dm-p]
[a-z&&[def]]	a 到 z 和 d、e、或 f (交集)
[a-z&&[^bc]]	a 到 z, 除 b 和 c 外 (相减)。也可指定为 [ad-z]
[a-z&&[^m-p]]	a 到 z, 而非 m 到 p (相减)。也可指定为 [a-lq-z]

预定义字符类

预定义字符类	匹配项
.	任意字符 (可能或不可能与行终止符匹配)
\d	任意数字: [0-9]
\D	非数字: [^0-9]
\s	空白字符: [\t\n\x0B\f\r]
\S	非空白字符: [^\s]
\w	单词字符: [a-zA-Z_0-9]
\W	非单词字符: [^\w]

边界匹配

边界匹配符	匹配项
<code>^</code>	行首
<code>\$</code>	行尾
<code>\b</code>	单词边界
<code>\B</code>	非单词边界
<code>\A</code>	输入的开头
<code>\Z</code>	除最后终止符外（如果有），输入的结尾
<code>\z</code>	输入的结尾

脚本编写语言

脚本编写语言概述

IBM® SPSS® Modeler 脚本编写语言的组成:

- 引用节点、流、工程、输出和其他 SPSS Modeler 对象的格式。
- 可用于操作这些对象的一组脚本编写语句或命令。
- 设置变量、参数和其他对象的值的脚本编写表达语言。
- 注释、连接符和文字文本块的支持。

本节说明脚本编译语言的基本语法。具体属性和命令的有关信息则在随后的章节中提供。

脚本编写语法

为了使语法分析过程更加清晰，在 IBM® SPSS® Modeler 中编写脚本时应遵循以下规则:

- 变量名称，比如 `income` 或 `referrerID`，不能加引号。
- 变量名称，比如 `^mystream`，在引用值已设定的现有变量时以插入符号 (^) 开头。声明或设置变量值时不使用插入符号。 [有关详细信息，请参阅第 19 页码引用节点。](#)
- 对会话、流和超节点参数的引用，比如 `'$P-Maxvalue'`，应使用单引号。
- 如果使用双引号，则表达式将被当作字符串文字，例如，`"Web graph of BP and Drug"`。如果不注意区分使用单引号和双引号，则可能导致出现意外的结果，例如，`"$P-Maxvalue"` 表示一个字符串，而不是参数中保存的值。
- 文件名应加双引号，比如，`"druglearn.str"`。
- 节点名称可以加单引号，也可以不加引号，比如 `databasenode` 或 `Na_to_K`。注意：如果名称包含空格或特殊字符，则必须加引号。但不能在脚本中使用以数字开头的节点名称，比如 `'2a_referrerID'`。
- 标志属性应使用 `true` 和 `false` 值（字母为小写）来进行读取和设置。设置值时也可识别如下变异值：`Off`、`OFF`、`off`、`No`、`NO`、`no`、`n`、`N`、`f`、`F`、`False`、`FALSE` 或 `0`，但在某些情况下读取属性值时会出错。所有其他值均视为 `true`。使用小写 `true` 和 `false` 时保持一致将可以避免混淆。
- 文字字符串或含有换行符、空格、单引号或双引号的块可以用三引号将内容引起来。 [有关详细信息，请参阅第 26 页码文字文本块。](#)
- CLEM 表达式，比如，`"Age >= 55"`，应加双引号。例如：

```
set :derivnode.flag_expr = "Age >= 55"
```
- 如果在 CLEM 表达式内使用引号，则请务必在引号前加反斜线 (\)，例如：

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

虽然并不是所有实例均要求严格遵守这些规定，但为了提高脚本的清晰性，最好予以遵守。脚本检查器在所有脚本编译对话框中均可以找到，它们可以标记出模糊的语法。

引用节点

脚本中节点的引用有若干种方法：

- 可按照名称指定节点，例如，`DRUG1n`。可使用类型来限定名称，例如，`Drug:neuralnetworknode` 指的是名称为 `Drug` 的神经网络节点，而不是任何其他类型的节点。
- 只能按照类型指定节点，例如，`:neuralnetworknode` 指的是所有神经网络节点。可以使用任何有效的节点类型，例如，`sampleneode`、`neuralnetworknode` 和 `kmeansnode`。`node` 后缀为可选并且可遗漏，但最好在名称中包含后缀，因为这样才能更轻松地识别脚本中的错误。
- 可按照节点的唯一 ID 来引用各个节点，唯一 ID 显示在各个节点的“注解”选项卡上。ID 前应使用“@”符号，例如，`@id5E5GJK23L.custom_name = "My Node"`。有关详细信息，请参阅第 5 章中的注解中的 IBM SPSS Modeler 15 用户指南。

已生成模型。 同样的规则适用于已生成模型节点。可使用管理器窗口中“已生成模型”选项卡上显示的节点名称，也可以按照类型引用已生成模型节点。注意：用于引用管理器已生成模型的名称与为了评分而添加到流的模型名称互不相同（后者使用“apply”前缀）。有关详细信息，请参阅第 39 页码第 4 章中的模型块名称。

用变量引用节点

可使用插入符（`^`）语法将提供的节点名称和类型用作局部脚本变量的值。例如，当需要节点名称时，`^n` 表示节点名称保存在变量 `n` 中，而 `Drug:^t` 则表示名称为 `Drug` 节点的节点类型保存在变量 `t` 中。

节点引用可保存在局部脚本变量中（使用 `var` 语句进行声明），但不保存在流、会话或超节点参数中。为了保证节点引用的明确性，应在创建节点时向变量分配一个唯一性节点 ID。

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

- 第一行创建名称为 `x` 的变量。
- 第二行则创建新的类型节点并将对节点的引用保存到 `x` 中。注意：`x` 保存对节点本身的引用，而不是保存节点名称。
- 第三行则将节点的 `custom_name` 属性的值设置为 `"My Node"`。插入符用于表示 `x` 是变量名称，而不是节点名称。（如果没有该插入符，则系统将查找名称为 `x` 的节点。因为 `var` 及其它命令的对象只能是变量，所以声明或设置变量时不需要插入符。但第三行中，`x` 在逻辑上可以是个节点名称，而不是变量，因此需要用插入符将两者区分开来。）

常见陷阱之一是并未事先声明变量就试图将对节点的引用保存的该变量中。

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

此示例中，`SET` 试图将 `x` 创建为流、会话或超节点参数，而不是创建为变量，并返回错误，这是因为对节点的引用无法保存在参数中。

按照 ID 引用节点

也可以将唯一性节点 ID 保存在变量中。例如：

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

循环通过流中节点。 也可以使用 `stream.nodes` 属性来返回流中所有节点的列表，然后迭代整个列表，以访问各个节点。有关详细信息，请参阅第 61 页码第 6 章中的流报告。

示例

NAME:TYPE

NAME 是节点名称，**TYPE** 则是节点类型。最低要求是必须含有 **NAME** 或 **TYPE**。即可以省略一个，但不能两者都省略。例如，以下命令将在名称为 `drug1n` 的现有可变文件节点和现有散点图节点（新节点不使用分号）之间的创建一个新的导出节点：

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

也可以在 **NAME** 或 **TYPE** 前添加插入符 (^)，以表示参数的存在，例如：

Drug:^t

此引用表示名称为 `Drug` 的节点，其中 `t` 是指定节点类型的参数。例如，如果 `^t` 的值为 `c50node`，则上述引用可以转换为：

Drug:c50node

同样，参数也可以用于节点名称。例如，以下项可以用于要求有节点名称的上下文中：

```
^n:derivenode
^n
```

检索对象

`get` 命令返回对流、节点或输出对象的引用，这样就可以用脚本来操作这些对象。例如：

```
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"
```

设置当前对象

以下特殊变量可用于引用当前对象：

- node
- stream
- output
- project

除了 `project` 之外，这些特殊变量还可进行重置，以更改当前上下文。这些变量与其他脚本编译变量不同，因为它们已被预先确定，所以无需先用 `var` 命令进行声明。

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

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

由于这些特殊变量与它们引用的对象的名称互相匹配，因而某些情况下变量和对象的区别人可能变得模糊，导致使用时产生微妙的差异。有关详细信息，请参阅第 30 页码第 4 章中的 `set` 命令。

注释

向特殊变量分配错误类型的值（比如将节点对象设置为变量 `stream`）将导致运行时错误。

如果可使用特殊变量，则任何变量均可以使用。例如，可使用以下命令来执行当前流的保存：

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

以下命令同样有效：

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

其中 `my_stream` 已被分配了一个流值。

打开流和其他对象

独立脚本中，可通过指定文件的文件名和位置，打开流，例如：

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

其他类型的对象可使用 `load` 命令打开，例如：

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

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

Open stream 与 load stream。 `load stream` 命令将指定的流添加到工作区，而不清除当前流中的节点。此命令在以前版本中使用更加广泛，如今大部分已被打开节点、管理节点和在多个流之间复制节点的功能所取代。

使用多个流

除了通过文件系统或 IBM® SPSS® Collaboration and Deployment Services Repository 对流进行访问的命令（`open`、`load` 和 `retrieve`），大多数脚本编写命令都自动适用于当前流。但有时可能想在独立脚本中打开和操作来自同一脚本的多个流。要完成此操作，可以设置对任何一打开流的引用，或使用 `with... endwith` 命令来暂时重新分配当前流。

例如，要关闭当前流以外的流，可使用 `get stream` 命令来引用所需要的流：

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

此脚本将把特殊变量流重新分配到流 `druglearn`（实际上使它变成当前流），然后关闭流。

此外，也可以使用 `with stream` 语句临时重新分配当前流，例如：

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

以上语句执行 `create` 操作，并在执行流脚本时将指定的流设置为当前流。执行每条语句后，原始当前流即被恢复。也可以包含条件语句和循环结构，例如：

```
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
```

上述语句将把循环中的所有表达式的当前流设置为 `STREAM`，并在已完成循环时恢复原始值。

局部脚本变量

局部脚本变量用 `var` 命令进行声明，并且只设置当前脚本的局部脚本变量。变量与参数不同，可以为会话、流或超节点设置变量，并且变量只能包含字符串或数字。

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

引用现有流时，务必在参数名称前使用插入符（`^`）。例如，以上脚本中：

- 第一行声明变量。
- 第二行设置变量值。

- 第三行重命名变量引用的节点（而不是变量本身）。插入符表示 `^my_node` 是变量名称，而不是节点的文字名称。（如果没有该插入符，则 `rename` 命令将查找名称为 `my_node` 的节点。因为 `var` 命令的对象只能是变量，因而第一行和第二行不需要插入符。（只有在引用已设置变量时，才能使用插入符，此时如果删除插入符，将产生模糊的引用。）
- 解析变量引用时，将先搜索局部变量列表，再搜索会话、流或超节点参数列表。例如，如果变量 `x` 作为局部变量和会话参数存在，在脚本编译语句中使用 `'$P-X'` 语法将确保使用会话参数，而不是局部变量。

注意：在实际应用的角度，如果不首先使用 `var` 命令声明的情况下设置变量，则会根据当前脚本的上下文创建流、会话或超节点参数。例如，以下编码将创建名称为 `z` 的局部脚本变量并将变量值设置为 `[1 2 3]`：

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

如果省略了 `var` 命令（并假定未存在该名称的变量或节点），则将 `z` 创建为参数，而不是变量。

流、会话和超节点参数

可以定义用于 CLEM 表达式和脚本编写的参数。这些参数在效果上相当于用户定义的变量，这些变量可以在当前流、会话或超节点中保存和保留，并且可以从用户界面和脚本编写过程中访问。例如，如果保存流，则为此流设置的所有参数也将保存。（此特点使这些参数区别于仅在脚本中使用和声明的本地脚本变量。）通常，参数在脚本编写中作为 CLEM 表达式的一部分使用，而此表达式中的参数值可在脚本中指定。

参数的范围取决于设置参数的位置：

- 流参数可以在流脚本中或在流属性对话框中设置，且可用于流中的所有节点。这些参数显示在表达式构建器的参数列表中。
- 会话参数可以在独立脚本中或在会话参数对话框中设置。这些会话参数可用于当前会话中使用的所有流（即在管理器窗格的“流”选项卡中列出的所有流）。

也可以为超节点设置参数，在这种情况下，这些参数仅对于该超节点中封装的节点可见。有关详细信息，请参阅第 9 章中的[定义超节点参数中的 IBM SPSS Modeler 15 源、过程和输出节点](#)。

在脚本中设置参数

可使用 `set` 命令和以下语法在脚本中设置参数：

```
set foodtype = pizza
```

如果不存在已在当前脚本中声明的名称为 `foodtype` 的节点或变量，则此命令将创建默认值为 `pizza` 的名称为 `foodtype` 的参数。

用户界面。 此外，也可以从“工具”菜单选择流属性或设置会话参数，通过用户界面设置或查看参数。这些对话框还可以用于指定无法通过脚本编译获得的其他选项，比如存储类型。有关详细信息，请参阅第 5 章中的设置流和会话参数中的 IBM SPSS Modeler 15 用户指南。

命令行。 也可以通过命令行设置参数，此情况下它们将被创建为会话参数。

引用脚本中的参数

引用以前创建的参数时，应为这些参数添加单引号，并加上前缀字符串 `$P`，例如，'`$P-minvalue`'。也可以仅引用参数名，比如 `minvalue`。参数值总是为字符串或数字。例如，可使用以下语法来引用 `foodtype` 参数并设置新值：

```
set foodtype = pasta
```

也可以在脚本中使用的 CLEM 表达式的环境内引用参数。以下是个示例脚本。它设置选择节点的属性，以将 `Age` 值大于名称为 `cutoff` 的流参数指定值的记录包括在内。该参数通过正确的 CLEM 语法（'`$P-cutoff`'）用于 CLEM 表达式中：

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

以上脚本使用名称为 `cutoff` 的流参数的默认值。可通过在选择节点规范上添加以下语法，指定新的参数值：

```
set cutoff = 50
```

所得出的脚本将选择所有 `Age` 值大于 50 的所有记录。

有关详细信息，请参阅第 7 章中的流、会话和超节点参数中的 IBM SPSS Modeler 15 用户指南。

控制脚本执行

脚本执行通常一句一句地处理语句。但是，可以通过使用条件 `if` 语句和多种 `for` 循环来覆盖此执行顺序，例如：

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

`for` 循环有多种格式，例如：

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

以上脚本使用列表顺序，为 LIST 中分配给 PARAMETER 的每个值执行一次 STATEMENTS。列表不带花括号，且其内容为常量。还存在其他若干可用格式。[有关详细信息，请参阅第 28 页码第 4 章中的一般脚本编写命令。](#)

脚本中的操作符

除了常用的 CLEM 操作符，还可以使用 “+” 和 “-” 操作符来操作局部脚本编译变量（用 var 命令声明）。+ 操作符将元素添加到列表，- 操作符则用于删除项。示例如下：

```
var z          # create a new local variable
set z = [1 2 3] # set it to the list containing 1, 2, and 3
set z = z + 4   # add an element; z now equals [1 2 3 4]
```

这些操作符不能与流、超节点或会话参数（用 set 命令在脚本中定义）一起使用或用在常规 CLEM 表达式之外（比如导出节点中的公式）。

CLEM 脚本中的表达式

可在 IBM® SPSS® Modeler 脚本中使用 CLEM 表达式、函数和操作符，但是脚本编写表达式不能含有对任何 @ 函数、日期/时间函数和位元操作的调用。另外，以下规则适用于脚本编写中的 CLEM 表达式：

- 指定的参数必须带有单引号并用 \$P- 作为前缀。
- CLEM 表达式必须包含在引号中。如果 CLEM 表达式本身含有加引号的字符串或字段名，则内嵌引号前必须带有反斜线 (\)。 [有关详细信息，请参阅第 18 页码脚本编写语法。](#)

可在脚本编译中使用全局值，比如 GLOBAL_MEAN(Age)，但不能在脚本编译环境中使用 @GLOBAL 函数本身。

以下是脚本编写中使用的 CLEM 表达式的示例：

```
set :balancenode.directives = [{1.3 "Age > 60"}]

set :fillernode.condition = "(Age > 60) and (BP = \"High\")"

set :derivenode.formula_expr = "substring(5, 1, Drug)"

set Flag:derivenode.flag_expr = "Drug = X"

set :selectnode.condition = "Age >= '$P-cutoff'"

set :derivenode.formula_expr = "Age - GLOBAL_MEAN(Age)"
```

插入注释和连接符

脚本编写中使用以下字符来表示注释和连接符。

字符	用法	示例
#	该散列符表示注释。行的剩余部分被忽略。	#这是单行注释。
\	以反斜线结尾的行表示语句会持续到下一行。	请参阅以下示例。
/*	序列 /* 表示注释的开头。直至出现 */ 注释结尾标记，这期间的所有内容都将被忽略。	请参阅以下示例。
"""	文字字符串或含有换行符、空格、单引号或双引号的块可以用三引号将内容引起来。有关详细信息，请参阅第 26 页 码文字文本块 。	

示例

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

```
#following is a multi-line statement
set :fixedfilenode.fields = [{"Age" 1 3}\
  {"Sex" 5 7} {"BP" 9 10} {"Cholesterol" 12 22}\
  {"Na" 24 25} {"K" 27 27} {"Drug" 29 32}]
```

文字文本块

含有空格、制表符和换行符的文字文本块可通过用三引号引起来，放在脚本中。引用块内的任何文本都将被保留为文字文本，包括空格、换行符、内嵌单引号和双引号。无需任何行连接或转义符。

例如，可使用此方法来将一组树生长指令内嵌在脚本中，如下文所示：

```
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
"""
```

此方法也可以用于路径和注解，例如：

```
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 将忽略开头文字标记之后的换行符。例如，以下示例相当于前个示例：

```
set :node.annotation = """
This node was built to help identify which of the following indicators
Etc...
"""
```

脚本编写命令

本节概述了可在 IBM® SPSS® Modeler 脚本中使用的命令，这些命令是按照对象类型组织的。有关脚本编写语言的详细信息，请参阅第 3 章。有关节点、流、工程和超节点属性的详细信息，请参阅第 9 章到第 22 章。

一般脚本编写命令

除非另有说明，否则下列命令可在所有独立脚本、流脚本和超节点脚本中使用。

execute_all

```
execute_all
```

执行当前流中的所有终端节点。

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

execute_script

```
execute_script
```

仅适用于独立脚本。执行与当前流相关联的流脚本。（仅限于独立脚本，因为用于其他脚本时可能会导致流脚本进行自我调用。）

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

exit

```
exit CODE
```

退出当前脚本。exit 代码可用于评估流或节点的脚本或条件，例如：

```
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

for...endfor 命令会根据条件循环一系列语句。该命令可以采用多种形式，所有这些形式均遵循相同的一般结构。

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

for PARAMETER in LIST。使用列表顺序，为 LIST 中分配给 PARAMETER 的每个值执行一次 STATEMENTS。例如，可以将多个字段的 Filter.include 属性设置为 true，可以按照下列方式完成：

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

for PARAMETER from N to M。针对 N 和 M 之间（包括 M 和 N）的每个整数都执行一次 STATEMENTS，例如：

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

for PARAMETER in_fields_to NODE。针对 NODE 上游一侧上的每个字段都执行一次 STATEMENTS。例如，下面的命令会将所有字段的 include 属性设置为 true，其中包括以前设置为 false 的字段：

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

注意：在节点可以具有多个相同名称输入字段的情况下（如合并节点或追加节点），此方法会返回下游字段的列表，而不是上游字段的列表，这样是为了避免任何可能导致不可预知结果的冲突。

for PARAMETER in_fields_at NODE。针对指定 NODE 中（或下游）的每个字段都执行一次 STATEMENTS。因此，如果节点是过滤节点的话，则只会包含通过的字段，并且该节点不应成为终端节点，因为不会返回任何字段。例如，与上面的情况相反，下面的脚本不会有任何效果，因为只会为那些已经设置为 true 的字段执行该循环：

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

for PARAMETER in_models。针对模型选项板中每个模型块都执行一次 STATEMENTS。例如，下列脚本会将选项板中的每个模型都插入到当前流中。（xpos 变量用于避免流工作区内发生一个节点堆叠在另一个节点上的情况。）

```

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

```

for PARAMETER in_streams. 仅适用于独立脚本。针对每个载入的流（在流选项板中列出）都执行一次 **STATEMENTS**。如果 **PARAMETER** 是特殊的变量流，则会为该循环中的 **STATEMENTS** 设置当前流。循环终止时将还原流的初始值。

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

```

if EXPR then
  STATEMENTS 1
else
  STATEMENTS 2
endif

```

指定的表达式为 `true` 时执行 **STATEMENTS 1**，指定的表达式为 `false` 时执行 **STATEMENTS 2**。`else` 子句是可选的。

```

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

```

set 命令

```

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

```

设置局部脚本变量、特殊变量、参数或属性的值。

设置变量

要设置局部脚本变量的值，首先使用 `var` 命令声明该变量，例如：

```

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

```

该变量的值可以是在脚本编译中有效的 CLEM 表达式、返回值的脚本命令（如 `load`、`create` 或 `get`），也可以是一个字面值。

```

set xpos = ^xpos + 50

var x
set x = create typenode

```

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

将特殊变量设置为引用对象

特殊变量 `node`、`stream`、`output` 和 `project` 用于引用每个上下文中的“当前”对象。除了 `project` 之外，这些特殊变量还可进行重置，以更改当前上下文。这些变量与其他脚本编译变量不同，因为它们已被预先确定，所以无需先用 `var` 命令进行声明。

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

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

这些变量都很有用，但在用法上会有细微区别，如下面的示例所示：

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

- 第一行会重置当前流，更精确地说，会设置特殊变量 `stream` 的值。（换句话说，`stream` 是一个变量，而不是该命令的一部分。）
- 第二行使用此变量为当前流设置一个属性（请参阅下面的内容详细了解有关属性的信息）。`^` 符号用于表明 `^stream` 是变量的名称，而不是对象（如节点）的名称。（如果没有该插入符，则 `set` 命令将查找名称为流的节点。）
- 最后两行则保存并关闭当前流。与前面一样，`stream` 是一个变量，但在这里没有使用 `^` 符号，因为本示例中使用的 `save` 和 `close` 命令只能应用于流。（通常只有当不使用 `^` 符号会引起模糊引用的情况下才使用该符号。）

引用当前工程。特殊变量 `project` 可用于引用当前工程（请参阅下面列出的设置工程属性的示例）。`project` 的值不能重置，因为任何时候一次都只能打开一个工程（因此为当前工程）。

设置参数

流、会话和超节点参数的设置方式可与变量相同，但不使用 `var` 命令。

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

注意：在实际应用的角度，如果 `set` 命令的对象与已声明变量、特殊变量或现有对象（如节点）的名称不匹配，则会创建参数。有关详细信息，请参阅第 23 页码第 3 章中的流、会话和超节点参数。

设置节点、流和工程属性

也可以为节点、流和工程设置属性，例如：

```

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

set ^stream.execute_method = "Script"

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

```

有关可用于节点、流和工程的属性完整列表，请参阅[属性参考信息第 98 页码](#)。

设置多个属性。您可以在一个操作中为节点或其他对象的属性赋予多个表达式。如果需要确定数据模型之前对节点进行多项更改，则使用此方法。用于设置多个属性的格式为：

```

set NODE {
    NODEPROPERTY1 = EXPRESSION1
    NODEPROPERTY2 = EXPRESSION2
}

```

例如：

```

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

set ^project {
    summary = "Initial modeling work on the latest data"
    ordering = NameAddedType
}

```

设置标志值（真或假）。读取或写入标志类型的属性时，值 `true` 和 `false` 应该为小写，例如：

```

set :variablefilenode.read_field_names = true

```

注意：设置值时也可识别如下变异值：`Off`、`OFF`、`off`、`No`、`NO`、`no`、`n`、`N`、`f`、`F`、`false`、`False`、`FALSE` 或 `0`，但在某些情况下读取属性值时会出错。所有其他值均视为 `true`。使用小写 `true` 和 `false` 时保持一致将可以避免混淆。

示例：设置节点属性

有很多节点特定的属性（有时称为通道参数）用于设置每个节点的用户界面对话框中的选项。例如，要创建流并为每个节点指定选项，您可以使用类似此处所显示的脚本。有关节点、流、工程和超节点属性的详细信息，请参阅第 9 章到第 22 章。

```

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

create tablenode at 400 100
create samplnode connected between :varfilenode and :tablenode
set :samplnode {
    max_size = 200
}

```

```
mode = "Include"
sample_type = "First"
}
create plotnode at 300 300
create derivenode connected between drug1n and :plotnode
set :derivenode {
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'
}
```

var 命令

```
var VARNAME
```

声明局部脚本变量。

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

变量与参数不同，可以为会话、流或超节点设置变量，并且变量只能包含字符串或数字。在实际应用的角度，如果不首先使用 **VAR** 命令声明的情况下设置变量，则会根据当前脚本的上下文创建流、会话或超节点参数。 [有关详细信息，请参阅第 22 页码第 3 章中的局部脚本变量。](#)

节点对象

下面的脚本编写命令可用于节点对象。

create NODE

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

创建指定类型的节点，例如：

```
create statisticsimportnode
```

还可以指定位置和连接选项：

```
create featureselectionnode at 400 100
create typenode between :statisticsimportnode and :featureselectionnode
```

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

您还可以使用变量来创建节点，以避免歧义。例如，在下面的示例中，会创建一个类型节点，并且引用变量 `x` 将设置为包含对该类型节点的引用。然后，您可以使用变量 `x` 返回由 `x` 引用的对象（本例中为类型节点），然后执行其他操作，如重命名、定位或连接新节点。

```
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
```

上面的示例会创建两个节点，对每个节点进行重命名，确定它们的位置，最后在流工作区中连接这两个节点。

图片 4-1
使用变量创建的节点



另外，特殊（预先定义的）变量 `node` 也可以与上面示例中的 `x` 和 `y` 变量相似的方式使用。这种情况下，不需要使用 `var` 命令声明该变量（因为已经预定义），生成的脚本可能读起来更容易一些。

```
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
```

注意：特殊变量（如 `node`）可重用以引用多个节点。使用 `set` 命令即可重置该变量引用的对象。有关详细信息，请参阅第 20 页码第 3 章中的设置当前对象。

复制节点。您还可以使用 `duplicate` 命令复制现有的节点。有关详细信息，请参阅第 35 页码 `duplicate NODE`。

connect NODE

```
connect NODE1 to NODE2
connect NODE1 between NODE2 and NODE3
```

将 `NODE1` 与指定的其他节点相连接。


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

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

delete NODE

```
delete NODE
```

从当前流中删除指定的节点。

```
delete :statisticsimportnode
```

```
delete DRUG1N:variablefilenode
```

disable NODE

```
disable NODE
```

从当前流禁用指定节点会在流执行期间忽略该节点。这种方式使您不必删除或绕过节点，并保持其连接到其余节点。您仍可以编辑节点设置，不过，除非重新启用节点，否则任何更改均不会生效。

```
disable :statisticsimportnode
```

```
disable DRUG1N:variablefilenode
```

disconnect NODE

```
disconnect NODE
```

```
disconnect NODE1 from NODE2
```

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

将指定的节点与所有其他节点（默认设置）或与指定的特定节点断开连接。

```
disconnect :typenode
```

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

duplicate NODE

```
duplicate NODE as NEWNAME
```

将一个新节点创建为指定节点的副本。还可以以绝对形式或相对形式指定位置。

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

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

enable NODE

```
enable NODE
```

在当前流中启用之前禁用的节点会在流执行期间包括该节点。如果您在节点已被禁用的情况下编辑了节点设置，则更改将立即生效。

```
enable :statisticsimportnode
```

```
enable DRUG1N:variablefilenode
```

execute NODE

```
execute NODE
```

执行指定的节点，例如：

```
execute :neuralnetworknode
```

如果该节点不是终端节点，执行则相当于从此处运行弹出菜单选项。

要执行当前流中的所有终端节点：

```
execute_all
```

仅适用于独立脚本。执行与当前流相关联的流脚本：

```
execute_script
```

注意：通过将流设置为当前流或者使用 `with` 命令，可以执行与不同流相关联的脚本。[有关详细信息，请参阅第 22 页码第 3 章中的使用多个流。](#)

export NODE as FILE

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

PMML 导出。要以 PMML 格式导出生成的模型：

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

SQL 导出。要以 SQL 格式导出生成的模型：

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

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

节点详细信息。要以 HTML 或文本格式导出节点详细信息：

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

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

节点概要。要以 HTML 或文本格式导出节点概要：

```
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
```

刷新流中指定节点或所有节点的高速缓存。如果某个给定节点的高速缓存未启用或者未满，此操作则不执行任何内容。

```
flush :mergenode
```

要执行当前流中的所有节点：

```
flush_all
```

get node NODE

```
get node NODE
```

获取现有节点的引用。这可能是一种用于确保节点非歧义引用的有用方式。

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

load node FILENAME

```
load node FILENAME
```

将保存的节点载入到当前流中。

```
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
```

以绝对或相对形式将节点定位在流工作区中。还可以选择指定连接选项：

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

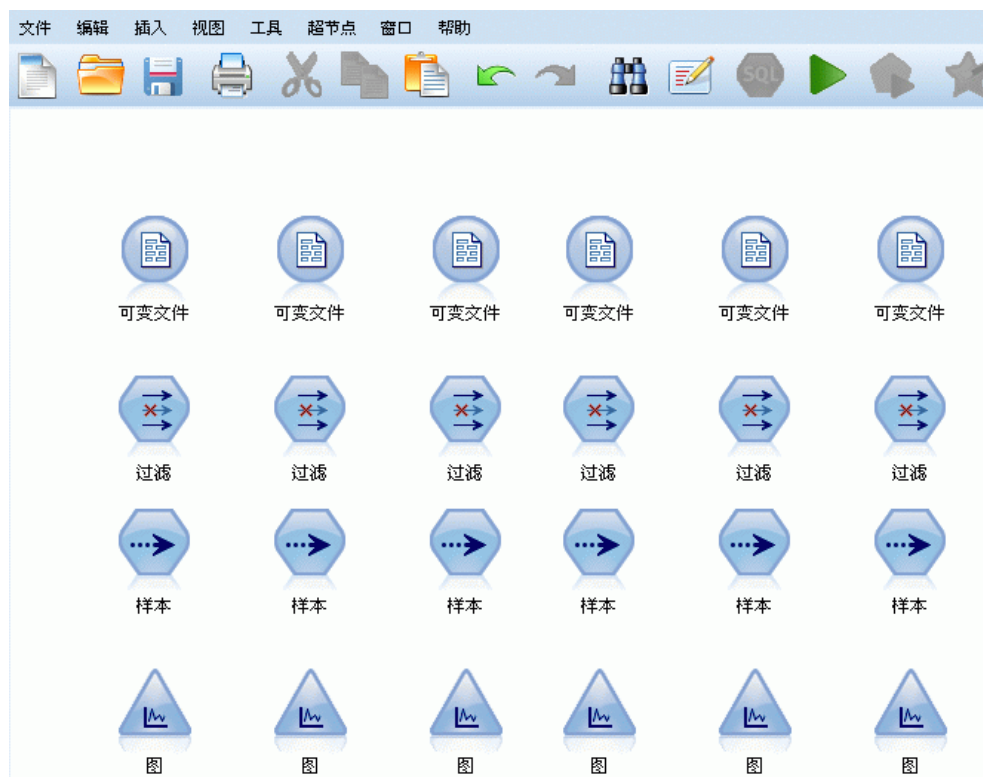
```
position Drug:net between DRUG2n and analysis
```

```
position :typenode connected between :variablefilenode and :tablenode
```

确定坐标位置

在流工作区中放置节点会使用一个不可见的 x-y 网格。您可以使用下图作为 x-y 网格坐标的参考。

图片 4-2
在 x-y 坐标指定的位置创建的节点



rename NODE as NEWNAME

```
rename NODE as NEWNAME
```

重命名指定的节点。

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

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

retrieve node REPOSITORY_PATH

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

从 IBM® SPSS® Collaboration and Deployment Services Repository 检索指定的节点。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

save node NODE as FILENAME

```
save node NODE as FILENAME
```

保存指定的节点。

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

store node NODE as REPOSITORY_PATH

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

将节点存储在 IBM® SPSS® Collaboration and Deployment Services Repository 中。
有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

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

模型对象

下面的脚本编写命令可用于模型对象。

模型块名称

模型块（也称为生成的模型）可按照类型引用，就好像节点和输出对象一样。下表列出了模型对象的引用名称。

请注意，这些名称专用于引用模型选项板（位于 IBM® SPSS® Modeler 窗口的右上角）中的模型块。要引用已经添加到流中进行评分的模型节点，则使用另外一套以 `apply...` 为前缀的名称。有关详细信息，请参阅第 224 页码第 17 章中的模型块节点属性。

例如，下面的脚本会将一个模型块添加到当前流中，将其与一个类型节点相连接，然后创建并执行一个表节点。请注意，用于插入选项板中模型的名称，与用于引用添加到流之后的“apply”模型节点的名称不同（`:featureselection` 与 `:applyfeatureselectionnode`）。

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

注意：这只是一个示例。在通常情况下，建议同时使用名称和类型来引用模型，以避免引起混淆（例如 `response_01:featureselection`）。

模型块名称（建模选项板）

模型名称	模型
anomalydetection	异常
Apriori	Apriori
autoclassifier	自动分类器
autocluster	自动聚类

模型名称	模型
autonumeric	自动数值
bayesnet	贝叶斯网络
c50	C5.0
carma	Carma
cart	C&R 树
chaid	CHAID
coxreg	Cox 回归
decisionlist	决策表
discriminant	判别
因子	主成分分析/因子
featureselection	特征选择
genlin	广义线性回归
kmeans	K-Means
knn	k-最近相邻元素
kohonen	Kohonen
linear	线性
logreg	Logistic 回归
neuralnetwork	神经网络
quest	QUEST
回归	线性回归
sequence	序列
slrm	自学响应模型
statisticsmodel	IBM® SPSS® Statistics模型
SVM	Support Vector Machine
timeseries	时间序列
两步聚类	两步

模型块名称（数据库建模选项板）

模型名称	模型
db2imassoc	IBM ISW 关联
db2imcluster	IBM ISW 聚类
db2imreg	IBM ISW 回归
db2imsequence	IBM ISW 序列
db2imtree	IBM ISW 决策树
msassoc	MS 关联规则
msbayes	MS Naive Bayes
mscluster	MS 聚类
mslogistic	MS Logistic 回归
msneuralnetwork	MS 神经网络

模型名称	模型
msregression	MS 线性回归
mssequencecluster	MS 序列聚类
mstimeseries	MS 时间序列
mstree	MS 决策树
oraabn	Oracle Adaptive Bayes
oraai	Oracle AI
oraapriori	Oracle Apriori
oradecisiontree	Oracle 决策树
oraglm	Oracle GLM
orakmeans	Oracle k-Means
oramdl	Oracle MDL
oranb	Oracle Naive Bayes
oranmf	Oracle NMF
oracluster	Oracle O-Cluster
orasvm	Oracle SVM

避免重复的模型名称

使用脚本对生成的模型进行操作时，务必注意：允许重复的模型名称可能会导致歧义引用。为了避免这种情况的发生，最好在脚本编译时要求对于生成的模型使用不同的名称。

要为重复模型名称设置选项：

- ▶ 从菜单中选择：
工具 > 用户选项
- ▶ 单击通知选项卡。
- ▶ 选择**替换原有模型**以限制生成的模型的重复命名。

如果存在模糊模型引用，SPSS Modeler 和 IBM SPSS Collaboration and Deployment Services 之间的脚本执行行为则会有所不同。SPSS Modeler 客户端包含“替换之前的模型”选项，可自动替换具有相同名称的模型（例如，脚本通过循环迭代每次皆产生不同的模型）。但是，当相同脚本在 IBM SPSS Collaboration and Deployment Services 中运行时，此选项不可用。要避免这种情况，您可以重命名每次迭代中产生的模型以避免对模型的模糊引用，或在循环结束之前清除现有的模型（例如，添加 `clear generated palette` 语句）。

delete model MODEL

```
delete model MODEL
```

从模型块选项板中删除指定的模型（或清除所有模型）。

```
delete model Drug
```

```
delete model Drug:c50
```

要删除当前脚本插入的最后一个模型：

```
delete last model
```

要使得这个 last 语句得以运行，必须在当前脚本执行中至少已经执行了一次 insert model 语句。

要从模型选项板中清除所有模型块：

```
clear generated palette
```

export model MODEL as FILE

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

PMML 导出。要以 PMML 格式导出生成的模型：

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

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

有关详细信息，请参阅第 10 章中的导入和导出 PMML 模型中的 IBM SPSS Modeler 15 用户指南。

SQL 导出。要以 SQL 格式导出生成的模型：

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

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

注意：SQL 导出仅适用于特定的模型类型。有关详细信息，请参阅第 3 章中的浏览模型块中的 IBM SPSS Modeler 15 建模节点。

模型详细信息。要以 HTML 或文本格式导出模型详细信息（浏览模型块时显示在“模型”选项卡中的详细信息）：

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

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

注意：这些格式不可用于没有“模型”选项卡的模型。

模型摘要。要以 HTML 或文本格式导出模型概要（浏览模型块时的“概要”选项卡）：

```
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
```


图片 4-3
导出为 HTML 的关联模型选项卡

	后项	前项	支持度 %	置信度 %
1	frozenmeal	beer and cannedveg	16.7	87.425
2	cannedveg	beer and frozenmeal	17.0	85.882
3	beer	frozenmeal and cannedveg	17.3	84.393
4	frozenmeal	beer	29.3	58.02
5	cannedveg	frozenmeal	30.2	57.285
6	frozenmeal	cannedveg	30.3	57.096
7	cannedveg	beer	29.3	56.997
8	beer	frozenmeal	30.2	56.291
9	beer	cannedveg	30.3	55.116
10	wine	confectionery	27.6	52.174
11	confectionery	wine	28.7	50.174

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
```

将模型添加到当前流。还可以指定位置和连接选项。

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

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

load model FILENAME

```
load model FILENAME
```

将保存的模型载入到模型选项板。

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

retrieve model REPOSITORY_PATH

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

从 IBM® SPSS® Collaboration and Deployment Services Repository 检索保存的模型。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

save model MODEL as FILENAME

```
save model MODEL as FILENAME
```

将指定的模型保存为一个生成的模型文件。

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

store model MODEL as REPOSITORY_PATH

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

将指定模型存储在 IBM® SPSS® Collaboration and Deployment Services Repository 中。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

扩展名 (*.gm) 是可选的，但关于它的使用必须在存储和检索模型时保持一致。例如，如果存储时只是存储为“Kohonen”，则需要按照同样的名称检索该模型。（换句话说，如果使用扩展名的话，它只是模型名称的一部分。）

流对象

下面的脚本编写命令可用于流对象。

create stream DEFAULT_FILENAME

```
create stream DEFAULT_FILENAME
```

仅适用于独立脚本。在内存中创建一个指定名称的新流。流不自动保存。

```
create stream 'Druglearn'
```

close STREAM

```
close STREAM
```

仅适用于独立脚本。关闭指定的流。

要关闭当前流，请全部使用小写字母键入命令，如下所示：

```
close stream
```

独立脚本

如果对多个流操作，务必注意：**stream**（此列中为小写形式）实际上是一个用以引用当前流的特殊变量。要关闭其他流，此变量的值则可以重置：

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

另外，还可以指定引用流的任何声明变量，例如：

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

最后，使用 **with stream** 命令可以临时重新分配当前流：

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

clear stream

```
clear stream
```

删除当前流中的所有节点。

get stream STREAM

```
get stream STREAM
```

仅适用于独立脚本。用于获取指定流的引用，该引用可以赋予局部变量（或者特殊变量 **stream**）。指定的流必须已经打开。

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

load stream FILENAME

```
load stream FILENAME
```

仅适用于独立脚本。将指定的流添加到工作区，而不清除当前流中的节点。

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

Open stream 与 load stream。 **load stream** 命令将指定的流添加到工作区，而不清除当前流中的节点。此命令在以前版本的 IBM® SPSS® Modeler 中使用更加广泛，在新版本中大部分已被打开节点、管理节点和在多个流之间复制节点的功能所取代。

open stream FILENAME

```
open stream FILENAME
```

仅适用于独立脚本。打开指定的流。

```
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)]
```

从 IBM® SPSS® Collaboration and Deployment Services Repository 检索指定的流。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

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

save STREAM as FILENAME

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

要保存当前流的更改（假设它以前已经保存），请全部使用小写字母键入该命令，如下所示：

```
save stream
```

要第一次使用新文件名保存流：

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

独立脚本

如果在独立脚本中对多个流进行操作，务必注意：**stream**（为上面所示的小写形式）实际上是一个用以引用当前流的特殊变量。要保存其他流，此变量的值则可以重置：

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

另外，还可以指定引用流的任何声明变量，例如：

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

最后，使用 **with stream** 命令可以临时重新分配当前流：

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

有关详细信息，请参阅第 22 页码第 3 章中的使用多个流。

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"
```

将当前流存储在 IBM® SPSS® Collaboration and Deployment Services Repository 中。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

独立脚本

如果在独立脚本中对多个流进行操作，务必注意：**stream**（为上面所示的小写形式）实际上是一个用以引用当前流的特殊变量。要存储其他流，此变量的值则可以重置：

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

另外，还可以指定引用流的任何声明变量，也可以使用 **with stream** 命令临时重新分配当前流。

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

with stream STREAM

```
with stream STREAM  
  STATEMENTS  
endwith
```

仅适用于独立脚本。执行 **STATEMENTS**，并将指定的 **STREAM** 设置为当前流。执行流之后将立即恢复原始的当前流。

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

工程对象

下面的脚本编译命令可用于工程对象。

扩展名 (*.cpj) 是可选的，但关于它的使用必须在存储和检索工程时保持一致。

execute_project

```
execute_project
```

生成当前工程的默认报告。

load project FILENAME

```
load project FILENAME
```

打开指定的工程。

```
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}
```

从 IBM® SPSS® Collaboration and Deployment Services Repository 检索工程。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

save project as FILENAME

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

保存当前工程。

store project as REPOSITORY_PATH

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

将当前工程存储在 IBM® SPSS® Collaboration and Deployment Services Repository 中。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

状态对象

使用 `load state` 命令可以载入保存的状态。

`load state FILENAME`

```
load state FILENAME
```

载入指定的状态。

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

结果对象

使用 `value` 命令可以访问结果。

`value RESULT`

```
value RESULT at ROW COLUMN
```

终端节点包括一个名为 `output` 的只读参数，通过该参数可以访问最近生成的对象。对于在行和列中生成表格输出的节点，此参数可以实现访问指定单元格的值，例如：

```
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
```

行和列是与 1 之间的偏移量。如果输出对象不存在，则返回错误。

结果对象属性

下列属性对于在行和列中包括数据的结果对象（如表和矩阵结果）是通用的：

属性	描述
<code>row_count</code>	返回数据中的行数。
<code>column_count</code>	返回数据中的列数。

文件对象

下面的脚本编写命令可用于文件对象。

`close FILE`

```
close FILE
```

上面的语句会关闭指定的文件。

open FILE

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

上面的语句会打开指定的文件。

- **create**。如果文件不存在就创建该文件，如果存在则重写该文件。
- **append**。追加到现有文件中。如果文件不存在则生成错误。

此语句会返回打开文件的文件句柄。

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

write FILE

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

上面的表达式会将文本表达式写入文件。第一个语句会按照原样写入文本，而第二个语句还会在写入表达式之后写入一个新行。如果 **FILE** 不是一个打开文件对象，会生成错误。

```
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
```

输出对象

下面的脚本编写命令可用于输出对象。

输出类型名称

下表列出了所有的输出对象类型和创建它们的节点。有关每种输出类型可用的导出格式完整列表，请参阅创建该输出类型的节点的属性说明，位于第 15 章，图形节点属性和第 19 章，输出节点属性中。

输出对象类型	节点
analysisoutput	分析
collectionoutput	收集

输出对象类型	节点
dataauditoutput	数据审核
distributionoutput	分布
evaluationoutput	评估
histogramoutput	直方图
matrixoutput	矩阵
meansoutput	均值
multiplotoutput	多重散点图
plotoutput	图
qualityoutput	质量
reportdocumentoutput	此对象类型不属于节点；它是工程报告创建的输出
reportoutput	报告
statisticsprocedureoutput	Statistics输出
statisticsoutput	Statistics
tableoutput	表
timeplotoutput	时间散点图
weboutput	Web

delete output OUTPUT

```
delete output OUTPUT
```

从管理器选项板中删除指定的输出。例如：

```
delete output :statisticsoutput
```

要从管理器选项板中删除所有输出项目：

```
clear outputs
```

export output OUTPUT

```
export output OUTPUT as FILE format FORMAT
```

以指定的格式导出输出。请注意，可用格式取决于输出类型，但在浏览指定输出时可以从“导出”菜单上找到这些可用格式。

```
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
```

获取指定输出的引用。例如，可以使用一个循环获取一系列输出对象并将其依次导出。

```
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
```

载入指定的输出。

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

retrieve output REPOSITORY_PATH

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

从 IBM® SPSS® Collaboration and Deployment Services Repository 检索指定的输出。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

save output OUTPUT as FILENAME

```
save output as FILENAME
```

保存指定的输出。

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

store output OUTPUT as REPOSITORY_PATH

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

将指定输出存储在 IBM® SPSS® Collaboration and Deployment Services Repository 中。有关详细信息，请参阅第 54 页码第 5 章中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。

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

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

脚本编写技巧

本章简要介绍使用脚本的技巧和方法，包括修改流执行、在脚本中采用加密密码以及访问 IBM® SPSS® Collaboration and Deployment Services Repository 中的对象等。

修改流执行

运行流时，将按默认情形下的优化顺序来执行其终端节点。某些情况下，您可能更喜欢以其他顺序来执行。要修改流的执行顺序，可以在流属性对话框的“脚本”选项卡上完成以下步骤：

- ▶ 打开一个空脚本。
- ▶ 单击工具栏上的追加默认脚本按钮来添加默认流脚本。
- ▶ 将默认流脚本中语句的顺序更改为您希望的执行顺序。

对节点执行循环

可以将 `for` 循环与 `^stream.nodes` 属性相结合对流中的所有节点进行循环。例如，下面的脚本就是对所有节点进行循环并将过滤节点中的字段名更改为大写。

此脚本可在含有过滤节点的任何流中使用，即便实际上并不过滤任何字段。只需添加传递所有字段的过滤节点即可在整个面板上将字段名称更改为大写。

```
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
```

该脚本根据 `^stream.nodes` 属性的返回值在当前流的所有节点中进行循环，并检查每个节点是否为过滤节点。如果是，则脚本将循环该节点中的每个字段，并用 `lowertoupper()` 函数将字段名更改为大写。

提示：要将字段名更改为小写，请使用 `uppertolower()` 函数。

访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象

注意：访问 IBM® SPSS® Collaboration and Deployment Services 存储库需要单独许可证。有关更多信息，请参阅 <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>。

如果已获得 IBM® SPSS® Collaboration and Deployment Services Repository 的使用许可，则可以使用脚本命令在存储库中存储、检索、锁定和解锁对象。通过此存储库，可以在企业应用程序、工具和解决方案环境中对数据挖掘模型和相关预测对象的生命周期进行管理。有关详细信息，请参阅第 9 章中的关于 IBM SPSS Collaboration and Deployment Services Repository 中的 IBM SPSS Modeler 15 用户指南。

连接到 IBM SPSS Collaboration and Deployment Services Repository

要访问存储库，必须首先通过 IBM® SPSS® Modeler 用户界面的“工具”菜单或使用命令行建立到该存储库的有效连接。（有关详细信息，请参阅第 65 页码第 7 章中的 IBM SPSS Collaboration and Deployment Services Repository 连接参数。）

存储和检索对象

在脚本中，可以使用 `retrieve` 和 `store` 命令来访问各种对象，其中包括流、模型、输出、节点和工程。命令语法如下：

```
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]
```

`REPOSITORY_PATH` 指出对象在存储库中的位置。路径必须用英文引号引起并以正斜杠作为分隔符。路径不区分大小写。

```
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"
```

另外，对象名中也可以包含扩展名，如 `.str` 或 `.gm`，但只要对象名一致，不一定必须有扩展名。例如，如果存储模型时未带扩展名，则检索时也不能带扩展名：

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

与此相对：

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

请注意，当检索对象时，除非指定版本或标签，否则始终返回对象的最新版本。检索节点对象时，节点将自动插入到当前流中。检索流对象时，必须使用独立脚本。不能从流脚本中检索流对象。

锁定和解锁对象

对于脚本，您可以锁定一个对象，以防止其他用户更新任一现有版本或新建版本。还可以解锁已锁定的对象。

锁定和解锁对象的语法为：

```
lock REPOSITORY_PATH
lock URI
```

```
unlock REPOSITORY_PATH
unlock URI
```

对于存储和检索对象，**REPOSITORY_PATH** 指出对象在存储库中的位置。路径必须用英文引号引起并以正斜杠作为分隔符。路径不区分大小写。

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

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

除此之外，还可以使用统一资源标识符（URI）而非存储库路径来给出对象的位置。URI 必须包含前缀 **spsscr:**，同时必须完全括在引号中。只有正斜杠可以作为路径分隔符，空格必须以编码形式出现。即在路径中以 **%20** 代替空格。URI 不区分大小写。示例如下：

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

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

注意，对象锁定适用于对象的所有版本 – 您无法锁定或解锁单个版本。

生成加密密码

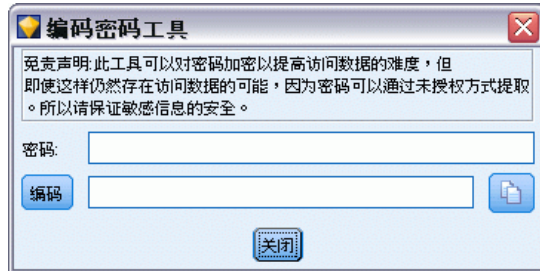
某些情况下，可能需要在脚本中包含密码，例如，您可能需要访问受密码保护的数据源。加密密码可用在：

- 数据库源和输出节点的节点属性
- 登录到服务器的命令行参数
- 存储在 **.par** 文件（由导出节点的“发布”选项卡生成的参数文件）中的数据库连接属性

通过此用户界面，可以使用一个工具根据 Blowfish 算法来生成加密密码（有关详细信息，请参阅 <http://www.schneier.com/blowfish.html>）。编码后，可以复制密码并将其存储到脚本文件和命令行参数中。加密密码存储在 **databasenode** 和 **databaseexportnode** 使用的节点属性 **epassword** 中。

- ▶ 要生成加密密码，请从“工具”菜单中选择：
编码密码...

图片 5-1
编码密码工具



- ▶ 在“密码”文本框中指定一个密码。
- ▶ 单击编码为自己的密码生成一个随机编码。
- ▶ 单击“复制”按钮将加密密码复制到剪贴板。
- ▶ 将此密码粘贴到所需的脚本或参数中。

脚本检查

通过单击“独立脚本”对话框工具栏上的红色检查按钮，可以快速检查所有类型脚本的语法。

图片 5-2
流脚本工具栏图标



脚本检查将就您编码中的错误发出警报并给出改进建议。要查看错误行，请单击该对话框下半部分的反馈。此时将以红色突出显示错误。

从命令行编写脚本

通过编写脚本可以运行通常在用户界面中执行的操作。启动 IBM® SPSS® Modeler 时，只需在命令行中指定和运行一个独立流。例如：

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

`-script` 标记会载入指定脚本，而 `-execute` 标记则会执行该脚本文件中的所有命令。

与早期版本的兼容性

在以前版本的 IBM® SPSS® Modeler 中创建的脚本通常应该无需更改就可以在当前版本中运行。不过，模型块现在可以自动插入到流中（此为默认设置），并可替代或补充流中此类型的现有模型块。实际发生的行为取决于将模型添加到流中和替换原有模型选项（工

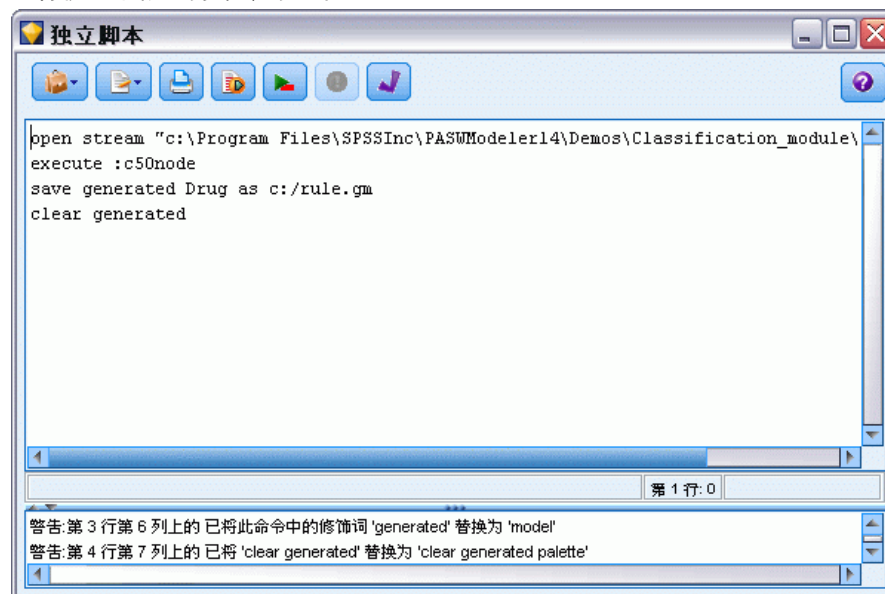
具 > 选项 > 用户选项 > 通知) 的设置。例如，您可能需要修改以前版本中的脚本，在该版本中模型块替换是通过删除现有模型块并插入新的模型块来完成。

在当前版本中创建的脚本在以前的版本中可能无法正常运行。

如果在旧版本中创建的脚本使用了已被替换（或不被支持）的命令，则使用旧形式命令的脚本仍然会得到支持，但将显示一条警告消息。例如，旧的 `generated` 关键字已由 `model` 替换，同时 `clear generated` 已由 `clear generated palette` 替换。沿用旧形式的脚本仍然可以运行，但将显示一条警告消息。

图片 5-3

运行使用不被支持命令的脚本



脚本编写示例

本节提供了大量示例演示如何在 IBM® SPSS® Modeler 中使用脚本。

类型节点报告

此脚本可创建有关当前流中的字段的 HTML 报告列表信息。此脚本可用于具有已实例化类型节点的任何流，且可以很容易地将其范围扩大到包括其他属性或节点。

- 使用标准 HTML 标记格式化标准浏览器中显示的结果。
- 使用 IBM® SPSS® Modeler 类型节点访问每个字段的属性。可以很便利地扩充脚本内容以列出“类型”节点中显示的其他属性，例如缺失值或字段角色。[有关详细信息，请参阅第 152 页码第 14 章中的 typenode 属性。](#)
- SPSS Modeler 脚本编写命令可用于将输出写入文件并在字段间循环以访问每个字段的属性。[有关详细信息，请参阅第 28 页码第 4 章中的脚本编写命令。](#)

图片 6-1
类型节点报告示例脚本

```
# 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>Type node information from SPSS Modeler</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border=\"1\">"
writeln myreport "<tr bgcolor=\"COCOCO\">"
writeln myreport "<td>Field</td><td>Type</td><td>Values</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>"
endifor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport

```

创建输出文件

首先，此脚本将创建一个新的 HTML 文件，并添加所需的标记以创建表，此表的标题行中列出了列标题Field、Type和Values。（每个 <td> </td> 标记对都会在表的行中创建一个单元格。）这些列将基于类型节点的属性应用于每个字段。

```

# 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>Type node information from SPSS Modeler</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border=\"1\">"
writeln myreport "<tr bgcolor=\"COCOCO\">"
writeln myreport "<td>Field</td><td>Type</td><td>Values</td>"
writeln myreport "</tr>"

```

在字段间循环

接下来，此脚本将在“类型”节点的所有字段间循环，并为每个字段添加一行，列出字段的名称和类型。

```

# 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>"
```

连续字段和标志字段的值

对于连续（数值范围）字段，`typenode.values` 属性将以格式 `[0.500517, 0.899774]`（显示在表中）返回最低值和最高值。对于标志字段，将以类似的格式显示真/假值。

```
# 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
```

名义字段的值

对于名义字段，`typenode.values` 属性将返回已定义值的完整列表。脚本将在此列表中循环，依次为每个字段插入每个值，并在每个值之间以换行符（`
` 标记）相间隔。

```
# 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
```

关闭文件

最后，此脚本将结束行，结束 `<table>`、`<body>` 和 `<html>` 标记，并结束输出文件。

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

流报告

此脚本可创建一个 HTML 报告，其中列出了当前流的每个节点的名称、类型和注解。除了创建 HTML 文件及访问节点和流属性的基本功能外，此脚本还说明了如何创建循环用以执行流中的每个节点的特定语句集。此脚本可在任何流中使用。

图片 6-2
流报告示例脚本

```
# 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
```

创建报告

首先，此脚本将创建一个具有 <BODY> 和 <HEAD> 元素的新 HTML 文件。^stream.name 属性将返回当前流的名称，并将此名称插入标题中。>> 运算符用于连接字符串。

```
# 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>"
```

然后，此脚本将创建一个具有标题行的 HTML 表，此标题行列出了列标题 Node Name、Type 和 Annotation。（每个 <td></td> 标记对都会在表的行中创建一个单元格。）

```
#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>"
```

接下来，此脚本将在当前流的所有节点间循环。每个节点的表中都将添加一行，用于列出名称、类型和注解。将在注解之后插入一个不可视的非中断空格 () 以避免在未指定给定节点的注解的情况下创建一个空单元格。（空单元格将导致在显示表时出现意外的格式。）

```
# 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
```

最后，此脚本将添加必需的 HTML 标记以结束文档和结束文件。

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

命令行参数

调用软件

您可以使用操作系统的命令行来如下启动 IBM® SPSS® Modeler:

- ▶ 在安装了 IBM® SPSS® Modeler 的计算机上, 打开 DOS 或命令提示符窗口。
- ▶ 要采用互动模式启动 SPSS Modeler 界面, 请输入后接所需参数的 `modelerclient` 命令; 例如:

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

可用参数 (标记) 允许您连接到一个服务器、加载流、运行脚本或根据需要指定其他参数。

命令行参数的使用

您可以将命令行参数 (也称为**标记**) 附加到最初的 `modelerclient` 命令以更改对 IBM® SPSS® Modeler 的调用。

例如, 可以使用 `-server`、`-stream` 和 `-execute` 标记来连接到服务器, 然后载入并运行流, 如下所示:

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

请注意, 当运行本地客户机安装程序时, 不需要输入服务器连接参数。

可以用双引号一括起包含空格的参数值, 例如:

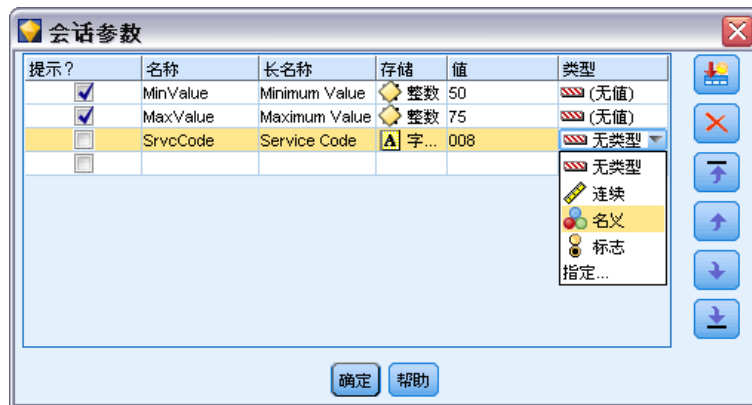
```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

还可以用此种方式执行 SPSS Modeler 状态和脚本, 但要分别使用 `-state` 和 `-script` 标记。

调试命令行参数

要调试命令行, 请使用 `modelerclient` 命令启动带有所需参数的 SPSS Modeler。这样可以验证命令是否可像期望中的那样运行。您还可以在“会话参数”对话框 (“工具”菜单 -> “设置会话参数”) 中对由命令行传递的参数进行确认。

图片 7-1
设置会话参数



组合多个参数

通过在文件名后使用 @ 标记，可以在调用时指定的命令文件中合并多个参数。这将使您可以缩短命令行调用，并且可以克服操作系统关于命令长度的限制。例如，以下启动命令使用了 <commandFileName> 的引用文件中的指定参数。

```
modelerclient @<commandFileName>
```

如果需用空格，则请用引号将命令文件的文件名和路径括起来，如下所示：

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

命令文件中可以包含在之前启动中单独指定的所有参数，每行一个参数。例如：

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

当写入和引用命令文件时，必须遵循以下限制：

- 每条命令占用一行。
- 不要在命令文件中嵌入 @CommandFile 参数。

服务器连接参数

-server 标记告诉 IBM® SPSS® Modeler 它可连接到公共服务器，同时标记 -hostname、-use_ssl、-port、-username、-password 和 -domain 用于告诉 SPSS Modeler 如何连接到公共服务器。如果未指定 -server 参数，则使用默认或本地服务器。

示例

连接到公共服务器：

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

连接到服务器群集：

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

请注意，连接到服务器群集需要通过在整个 IBM® SPSS® Collaboration and Deployment Services 中使用过程协调器，因此 `-cluster` 参数必须与存储库连接选项 (`spsscr_*`) 结合使用。有关详细信息，请参阅第 65 页码 [IBM SPSS Collaboration and Deployment Services Repository 连接参数](#)。

参数	行为 / 描述
<code>-server</code>	在服务器模式下运行 SPSS Modeler，使用标记 <code>-hostname</code> 、 <code>-port</code> 、 <code>-username</code> 、 <code>-password</code> 和 <code>-domain</code> 连接到公共服务器。
<code>-hostname <name></code>	服务器的主机名称。仅在服务器模式下可用。
<code>-use_ssl</code>	指定连接应采用 SSL（安全套接字层）。此标记为可选项，默认设置为不使用 SSL。
<code>-port <number></code>	指定服务器的端口号。仅在服务器模式下可用。
<code>-cluster <name></code>	指定指向服务器聚类（而不是已命名的服务器）的连接；此参数可用于替代 <code>hostname</code> 、 <code>port</code> 和 <code>use_ssl</code> 参数。名称为聚类名，或标识 IBM® SPSS® Collaboration and Deployment Services Repository 中聚类的唯一 URI。服务器聚类由 IBM SPSS Collaboration and Deployment Services 中的过程协调器管理。有关详细信息，请参阅第 65 页码 IBM SPSS Collaboration and Deployment Services Repository 连接参数 。
<code>-username <name></code>	登录服务器的用户名。仅在服务器模式下可用。
<code>-password <password></code>	登录服务器的密码。仅在服务器模式下可用。注意：如果未使用 <code>-password</code> 参数，系统将提示您使用密码。
<code>-epassword <encodedpasswordstring></code>	登录服务器的加密密码。仅在服务器模式下可用。注意：可以从 SPSS Modeler 应用程序的“工具”菜单中生成加密密码。
<code>-domain <name></code>	登录服务器的域名。仅在服务器模式下可用。
<code>-P <name>=<value></code>	用于设置启动参数。还可用于设置节点属性（通道参数）。

IBM SPSS Collaboration and Deployment Services Repository 连接参数

注意：访问 IBM® SPSS® Collaboration and Deployment Services 存储库需要单独许可证。有关更多信息，请参阅 <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>。

如果想通过命令行来存储或检索 IBM SPSS Collaboration and Deployment Services 中的对象，则必须指定一个指向该 IBM® SPSS® Collaboration and Deployment Services Repository 的有效连接。例如：

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

下表列出了可用于建立连接的参数：

参数	行为 / 描述
<code>-spsscr_hostname</code> <主机名或 IP 地址>	安装 IBM SPSS Collaboration and Deployment Services Repository 的服务器的 hostname 或 IP 地址。
<code>-spsscr_port</code> <number>	IBM SPSS Collaboration and Deployment Services Repository 接受连接的端口号（通常，默认值为 8080）。
<code>-spsscr_use_ssl</code>	指定连接应采用 SSL（安全套接字层）。此标记为可选项，默认设置为不使用 SSL。
<code>-spsscr_username</code> <name>	登录到 IBM SPSS Collaboration and Deployment Services Repository 的用户名。
<code>-spsscr_password</code> <password>	登录到 IBM SPSS Collaboration and Deployment Services Repository 的密码。
<code>-spsscr_epassword</code> <加密密码>	登录到 IBM SPSS Collaboration and Deployment Services Repository 的加密密码。
<code>-spsscr_domain</code> <name>	登录到 IBM SPSS Collaboration and Deployment Services Repository 所使用的域。此标记为可选项—除非您使用 LDAP 或 Active Directory 登录，否则请不要使用此项。—

系统参数

下表介绍可用于用户界面命令行调用的系统参数：

参数	行为 / 描述
<code>@</code> <commandFile>	@ 符号后跟文件名，此文件用于指定命令列表。当 <code>modelerclient</code> 遇到以 @ 开头的参数时，它将在该文件中对命令进行操作，就如同在命令行中一样。 有关详细信息，请参阅第 64 页码组合多个参数。
<code>-directory</code> <dir>	设置默认工作目录。在本地模式下，该目录将同时用于数据操作和输出。
<code>-server_directory</code> <dir>	为数据设置默认服务器目录。通过 <code>-directory</code> 标记指定的工作目录将用于输出。
<code>-execute</code>	在启动后执行启动时所载入的流、状态或脚本。如果在流或状态之外还载入了脚本，则脚本将单独执行。
<code>-stream</code> <流>	启动时载入指定的流。可以指定多个流，但是最后一个指定的流将被设置为当前流。
<code>-script</code> <script>	启动时载入指定的独立脚本。如下所述，除流或状态之外，此标记还可用于指定脚本，但在启动时仅可载入一个脚本。
<code>-model</code> <模型>	在启动时载入指定的已生成模型（.gm 格式的文件）。
<code>-state</code> <状态>	启动时载入指定的已保存状态。
<code>-project</code> <工程>	载入指定工程。在启动时仅可载入一个工程。
<code>-output</code> <输出>	在启动时载入已保存的输出项目（.cou 格式的文件）。
<code>-help</code>	显示命令行参数列表。指定此选项后，将忽略所有其它参数并显示帮助屏幕。
<code>-P</code> <name>=<value>	用于设置启动参数。还可用于设置节点属性（通道参数）。

注意：还可在用户界面中设置默认目录。要访问上述选项，请在“文件”菜单中选择设置工作目录或设置服务器目录。

载入多个文件

命令行模式下，您可以通过在启动时重复输入每个载入对象的相关参数来载入多个流、状态和输出。例如，要载入和运行两个称为 report.str 和 train.str 的流，您可以使用如下命令：

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

从 IBM SPSS Collaboration and Deployment Services Repository 加载对象

由于可从一个文件或 IBM® SPSS® Collaboration and Deployment Services Repository（如果获得许可）加载对象，文件名将使用前缀 spsscr:，也可以使用 file:（对于硬盘上的对象）来告诉 IBM® SPSS® Modeler 查找对象的地方。前缀可与以下标记配合使用：

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

您可以使用前缀创建 URI 以指定对象的位置—例如，-stream "spsscr:///folder_1/scoring_stream.str"。由于 spsscr: 前缀的存在，这需要在同一命令中指定到 IBM SPSS Collaboration and Deployment Services Repository 的有效连接。因此，完整的命令应形如以下的示例：

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

有关 IBM SPSS Collaboration and Deployment Services Repository 中对象的 URL 的详细信息，请参阅第 5 章第 54 页码中的访问 IBM SPSS Collaboration and Deployment Services Repository 中的对象。注意，在命令行中必须使用 URI。不支持像 REPOSITORY_PATH 这样的简单路径。（此种路径仅适用于脚本。）

形参的实参值

参数可用作在 IBM® SPSS® Modeler 的命令行执行期间的标记。在命令行参数中，-P 标记用于表示形式参数 -P <name>=<value>。

参数可以是：

- **简单参数**（或直接用在 CLEM 表达式中的参数）。
- **通道参数**，还可以是**节点属性**。此类参数可用于修改流中节点的设置。[有关详细信息，请参阅第 100 页码第 9 章中的节点属性概述。](#)
- **命令行参数**，用于更改对 SPSS Modeler 的调用。

例如，您可以提供数据源用户名和密码作为命令行标志，如下所示：

```
modelerclient -stream response.str -P:databasenode.datasource=["ORA 10gR2", user1, mypsw, true]
```

其格式与 `databasenode` 节点属性的 `datasource` 参数相同。有关详细信息，请参阅第 110 页码第 12 章中的 `databasenode` 属性。

CLEM 语言参考

CLEM参考概述

本节介绍表达式操作控制语言 (CLEM)，它是一个用来分析和操控 IBM® SPSS® Modeler 流中所使用数据的强大工具。可以在节点内使用 CLEM 来执行从评估条件或推导值到将数据插入报告等一系列任务。有关详细信息，请参阅第 7 章中的关于 CLEM 中的 IBM SPSS Modeler 15 用户指南。

部分 CLEM 语言还可在（用户界面中）编写脚本时使用。从而可使许多相同数据操作以自动化方式执行。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

CLEM 表达式由值、字段名、运行符和函数组成。使用正确的语法，可以创建各种功能强大的数据操作。有关详细信息，请参阅第 7 章中的 CLEM 示例中的 IBM SPSS Modeler 15 用户指南。

CLEM数据类型

CLEM 数据类型可由以下任意一项构成：

- 整数
- 实数
- 字符
- 字符串
- 列表
- 字段
- 日期/时间

引用规则

虽然 IBM® SPSS® Modeler 在确定 CLEM 表达式中使用的字段、值、参数和字符串时非常灵活，但在创建表达式时遵循以下一般规则将有利于您的实践操作：

- 字符串—编写字符串时始终用英文双引号将其括入，如 "Type 2"。也可以使用单引号，但容易与带引号字段混淆。
- 字段—只有需要将空格或其他特殊符号括入时才使用单引号，如 'Order Number'。加引号但数据集中未定义的字段会误读为字符串。
- 参数—使用参数时始终用英文单引号，如 '\$P-threshold'。
- 字符—始终使用反单引号 (``)，如 stripchar(`d`, "drugA")。

有关详细信息，请参阅第 7 章中的值和数据类型中的 IBM SPSS Modeler 15 用户指南。另外，以下主题详细介绍了这些规则。

整数

整数表示为十进制数字序列。可以选择在整数前加负号 (-) 来表示负数—例如 1234、999、-77。

CLEM 语言可以处理任意精度的整数。最大整数大小取决于您的平台。如果值太大而无法以整数字段显示，将字段类型更改为 **Real** 通常可以解决这一问题。

实数

实数 是指浮点数。实数的形式为一个或多个数字后跟一个小数点，小数点后再跟一个或多个数字。CLEM 实数为双精度数。

可以选择在实数前加负号 (-) 来表示负数 - - 例如 1.234、0.999、-77.001。可以用 <数字> e <指数> 形式以指数表示法表示实数—例如，1234.0e5、1.7e-2。当 IBM® SPSS® Modeler 应用程序从文件中读取数字字符串并自动将它们转换为数字时，它接受小数点之间无前导数字或小数点之后无数字的数字 - - 例如，999. 或 .11。不过，这些形式在 CLEM 表达式中是非法的。

注意：在 CLEM 表达式中引用实数时，无论当前流或环境的设置如何，都必须使用句点作为小数分隔符。例如，指定

```
Na > 0.6
```

而不是

```
Na > 0,6
```

即使在流属性对话框中选择了逗号作为小数符号，此规则也适用，它与代码语法应独立于任何特定环境或惯例的一般准则一致。

字符

字符（通常显示为 **CHAR**）通常在 CLEM 表达式中用于对字符串进行检验。例如，可以使用函数 `isuppercode` 来验证某个字符串的第一个字符是否为大写。下面的 CLEM 表达式使用一个字符来表示应对字符串的第一个字符进行这样的检验：

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

要在 CLEM 表达式中表示特定字符的代码（相对于位置），请使用 ``<character>`` 形式的反单引号，例如，`A`、`Z`。

注意：对于字段而言，没有 **CHAR** 存储类型，如果使用会产生 **CHAR** 存储类型的表达式派生或填充字段，则结果将转换为字符串。

字符串

一般而言，应将字符串用英文双引号括起。例如，字符串 "c35product2" 和 "referrerID"。要在字符串中表示特殊字符，请使用反斜杠 - - 例如，"\$65443"。（要表示反斜杠字符，应使用两个反斜杠 \\。）也可以使用英文单引号将字符串括

起，但这样会导致无法与带引号字段 ('referrerID') 相区别。有关详细信息，请参阅字符串函数中的 IBM SPSS Modeler 15 用户指南。

列表

列表是元素的有序排列，其中的元素可以是混合类型。列表以方括号 ([]) 括起。例如，列表 [1 2 4 16] 和 ["abc" "def"]。列表并不用作 IBM® SPSS® Modeler 字段的值。它们为函数（如 member 和 oneof）提供参数。

字段

CLEM 表达式中的名称，如若不是函数名称，则视其为字段名称。这些名称可以简单地写作 Power、val27、state_flag 等，但是，如果名称以数字开头或包含非字母字符，比如空格（下划线除外），请用单引号将其括起——例如，'Power Increase'、'2nd answer'、'#101'、'\$P-NextField'。

注意：加引号但数据集中未定义的字段会误读为字符串。

日期

日期将以在“流属性”对话框中指定的“基线”日期为依据进行计算。默认基线日期为 1900 年 1 月 1 日。有关详细信息，请参阅第 5 章中的设置流的常规选项中的 IBM SPSS Modeler 15 用户指南。

CLEM 语言支持以下日期格式。

格式	示例
DDMMYY	150163
MMDDYY	011563
YYMMDD	630115
YYYYMMDD	19630115
YYYYDDD	前四位数表示年，后三位数表示一年中的某一天—例如，2000032 表示 2000 年的第 32 天或 2000 年 2 月 1 日。
DAY	当前区域设置下一周中的某一天—例如，英语环境下的 Monday、Tuesday 等。
MONTH	当前区域设置下的月份—例如，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

格式	示例
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
q Q YYYY	日期由代表季度的数字 (1 - 4)、字母 Q 以及代表年的四位数字表示。例如, 25th Dec 2004 将表示为 4 Q 2004。
ww WK YYYY	前两位数字代表一年中的某周, 后跟字母 WK 以及代表年的四位数字。计算一年中的某周时, 假设这周的第一天是星期一且第一周至少有一天。

时间

CLEM 语言支持以下时间格式。

格式	示例
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

CLEM运算符

CLEM 语言提供了下列运算符。

运算	注释	优先级（参阅下一节）
or	用于两个 CLEM 表达式之间。如果其中一个表达式为真或两个表达式同时为真，则返回真值。	10
and	用于两个 CLEM 表达式之间。如果两个表达式同时为真，则返回真值。	9
=	用于任意两个可比项之间。若 ITEM1 等于项目 ITEM2，则返回值为真。	7
==	与 = 相同。	7
/=	用于任意两个可比项之间。若 ITEM1 不等于 ITEM2，则返回值为真。	7
/==	与 /= 相同。	7
>	用于任意两个可比项之间。若 ITEM1 完全大于 ITEM2，则返回值为真。	6
>=	用于任意两个可比项之间。若 ITEM1 大于或等于 ITEM2，则返回值为真。	6
<	用于任意两个可比项之间。如果 ITEM1 严格小于 ITEM2，则返回真值。	6
<=	用于任意两个可比项之间。若 ITEM1 小于或等于 ITEM2，则返回值为真。	6
&&=_0	用于两个整数之间。相当于布尔表达式 <code>INT1 && INT2 = 0</code> 。	6
&&/=_0	用于两个整数之间。相当于布尔表达式 <code>INT1 && INT2 /= 0</code> 。	6
+	将两个数字相加： <code>NUM1 + NUM2</code> 。	5
><	连接两个字符串，例如 <code>STRING1 >< STRING2</code> 。	5
-	将两个数字相减： <code>NUM1 - NUM2</code> 。也可以用在数字之前： <code>- NUM</code> 。	5
*	用于将两个数字相乘： <code>NUM1 * NUM2</code> 。	4
&&	用于两个整数之间。结果为 INT1 和 INT2 的位元“与”。	4
&&~~	用于两个整数之间。结果为 INT1 的位元“与”和 INT2 的位元补码。	4
	用于两个整数之间。结果为 INT1 和 INT2 的位元“或”。	4

运算	注释	优先级（参阅下一节）
~~	用于整数前。生成整数的位元补码。	4
/&	用于两个整数之间。结果为 INT1 和 INT2 的位元“异或”。	4
INT1 << N	用于两个整数之间。产生整数向左偏移 N 个位置的位元模式。	4
INT1 >> N	用于两个整数之间。产生整数向右偏移 N 个位置的位元模式。	4
/	用于将两个数相除：NUM1 / NUM2。	4
**	用于两个数值之间：BASE ** POWER。返回 BASE 的 POWER 次幂。	3
rem	用于两个整数之间：INT1 rem INT2。返回余数，INT1 - (INT1 div INT2) * INT2。	2
div	用于两个整数之间：INT1 div INT2。执行整数除。	2

运算符优先级

优先级决定复杂表达式（尤其是含有不止一个中缀运算符且未加括号的表达式）的解析顺序。例如，

$$3 + 4 * 5$$

解析为 $3 + (4 * 5)$ 而不是 $(3 + 4) * 5$ ，因为相对优先级规定 * 应先于 + 解析。CLEM 语言中的每个运算符均有一个与其关联的优先级值。此值越大，运算符在解析列表中的重要性越高，即它将先于优先级值比它大的其他运算符处理。

函数参考

下列 CLEM 函数可用于在 IBM® SPSS® Modeler 中处理数据。可以在各种对话框（如导出节点和设为标志节点）中以代码形式输入这些函数，也可以使用表达式构建器来创建有效的 CLEM 函数表达式，而后一种方式无需记住函数列表和字段名。

函数类型	描述
信息	用于深入了解字段值。例如，函数 <code>is_string</code> 针对类型为字符串的所有记录返回真值。
转换	用于构建新字段或转换存储类型。例如，函数 <code>to_timestamp</code> 会将选定字段转换为时间戳。
比较	用于字段值的相互比较或与指定字符串进行比较。例如， <code><=</code> 用来比较两个字段的值是否有一个更小或是相等。
逻辑	用来进行逻辑运算，例如， <code>if</code> 、 <code>then</code> 、 <code>else</code> 运算。
Numeric	用来进行数值计算，例如对字段值取自然对数。
三角法	用来进行三角计算，例如指定角度的反余弦。
Probability	返回各种分布的概率，例如，学生氏 t 分布中某个值将小于特定值的概率。
位元	用于以位元模式操作整数。

函数类型	描述
Random	用于随机选择项或生成 数值。
字符串	用于对字符串进行各种操作，例如 <code>stripchar</code> 就是用来删除指定字符。
SoundEx	用于在不知道字符串准确拼写的情况下根据某些字母的假设发音查找字符串。
日期和时间	用于对日期、时间和时间戳字段执行各种操作。
序列	用于深入了解数据集的记录序列或根据该序列进行操作。
全局量	用于访问由设置全局量节点创建的全局值。例如， <code>@MEAN</code> 用于引用某个字段在整个数据集中所有值的平均值。
空值和 Null 值	用于访问、标记或填充用户指定的空值或系统缺失值。例如， <code>@BLANK (FIELD)</code> 用于为存在空值的记录添加一个真值标志。
特殊字段	用于标示检查中的特定字段。例如，在派生多个字段时使用 <code>@FIELD</code> 。

函数说明中的约定

涉及函数中的项时，以下惯例通用于本指南。

惯例	描述
BOOL	布尔值或标志，例如真或假。
NUM、NUM1、NUM2	任意数字。
REAL、REAL1、REAL2	任意实数，例如 1.234 或 -77.01。
INT、INT1、INT2	任意整数，例如 1 或 -77。
CHAR	字符代码，例如 `A`。
STRING	字符串，例如 "referrerID"。
LIST	项目列表，例如 ["abc" "def"]。
ITEM	字段，例如 <code>Customer</code> 或 <code>extract_concept</code> 。
DATE	其值具有 DD-MON-YYYY 这样格式的日期字段，例如 <code>start_date</code> 。
TIME	其值具有 HHMMSS 这样格式的时间字段，例如 <code>power_flux</code> 。

本指南中的函数以函数、结果类型（整数、字符串等）和说明（如果有）各占一列的形式一一列举说明。例如，函数 `rem` 的说明如下。

函数	结果	描述
<code>INT1 rem INT2</code>	Number	返回 INT1 除以 INT2 所得的余数。例如， <code>INT1 - (INT1 div INT2) * INT2</code> 。

有关用法约定的详情（例如，如何在函数中列出项目或指定字符）将在其他文档中介绍。有关详细信息，请参阅CLEM数据类型中的IBM SPSS Modeler 15 用户指南。

信息函数

信息函数用于深入了解特定字段的值。它们通常用于派生标志字段。例如，可以使用 @BLANK 函数来创建一个标志字段，以指示选定字段的值为空值的记录。同样，可以使用存储类型函数（如 is_string）来检查某个字段的存储类型。

函数	结果	描述
@BLANK (FIELD)	布尔逻辑	根据上游类型节点或源节点（“类型”选项卡）中所设置的空值处理规则，对所有空值的记录返回真值。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
@NULL (ITEM)	布尔逻辑	为未定义值的所有记录返回真值。未定义值是系统 Null 值，在 IBM® SPSS® Modeler 中显示为 \$null\$。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
is_date (ITEM)	布尔逻辑	为所有日期类型的记录返回真值。
is_datetime (ITEM)	布尔逻辑	为所有日期、时间或时间戳类型的记录返回真值。
is_integer (ITEM)	布尔逻辑	为所有整数类型的记录返回真值。
is_number (ITEM)	布尔逻辑	为所有数值类型的记录返回真值。
is_real (ITEM)	布尔逻辑	为所有实数类型的记录返回真值。
is_string (ITEM)	布尔逻辑	为所有字符串类型的记录返回真值。
is_time (ITEM)	布尔逻辑	为所有时间类型的记录返回真值。
is_timestamp (ITEM)	布尔逻辑	为所有时间戳类型的记录返回真值。

转换函数

转换函数可用于构建新字段和转换现有文件的存储类型。例如，可通过将字符串连接在一起或分拆字符串来形成新字符串。要连接两个字符串，请使用运算符 <>。例如，如果字段 Site 的值为 "BRAMLEY"，则 "xx" <> Site 将返回 "xxBRAMLEY"。即使参数不是字符串，<> 的结果也始终是字符串。因此，如果字段 V1 为 3，字段 V2 为 5，则 V1 <> V2 将返回 "35"（字符串而非数字）。

转换函数（及其他要求特定类型输入（如日期或时间值）的函数）取决于“流选项”对话框中指定的当前格式。例如，如果想转换值为 Jan 2003、Feb 2003 等的字符串字段，请选择匹配的日期格式 MON YYYY 作为流的默认时间格式。有关详细信息，请参阅第 5 章中的设置流的常规选项中的 IBM SPSS Modeler 15 用户指南。

函数	结果	描述
ITEM1 <> ITEM2	字符串	连接两个字段的值，并返回结果字符串 ITEM1ITEM2。
to_integer (ITEM)	整数	将指定字段的存储值转换为整数。
to_real (ITEM)	实数	将指定字段的存储值转换为实数。
to_number (ITEM)	Number	将指定字段的存储值转换为数值。
to_string (ITEM)	字符串	将指定字段的存储值转换为字符串。
to_time (ITEM)	时间	将指定字段的存储值转换为时间。
to_date (ITEM)	日期	将指定字段的存储值转换为日期。

函数	结果	描述
to_timestamp(ITEM)	时间戳	将指定字段的存储值转换为时间戳。
to_datetime(ITEM)	日期时间	将指定字段的存储值转换为日期、时间或时间戳值。
datetime_date(ITEM)	日期	返回数字、字符串或时间戳的日期值。请注意，这是唯一允许您将数字（以秒数记）转换回日期的函数。如果 ITEM 为字符串，将以当前日期格式解析字符串来创建日期。只有“流属性”对话框中指定的日期格式正确，此函数方能成功执行。如果 ITEM 为数字，将解析为自基线日期（或纪元）以来的秒数。不足一天的部分将被截断。如果 ITEM 为时间戳，将返回时间戳的日期部分。如果 ITEM 为日期，将返回未更改的日期。

比较函数

比较函数用于字段值的相互比较或与指定字符串进行比较。例如，可以使用 = 来检查字符串是否相等。字符串相等验证示例：`Class = "class 1"`。

对数值比较来说，大于 表示离正无穷更近，小于 表示离负无穷更近。即，所有负数均小于任意正数。

函数	结果	描述
count_equal(ITEM1, LIST)	整数	返回字段列表中等于 ITEM1 的值的个数，如果 ITEM1 为 Null，则返回 Null。 有关详细信息，请参阅第 7 章中的多个字段的汇总中的 IBM SPSS Modeler 15 用户指南。
count_greater_than(ITEM1, LIST)	整数	返回字段列表中大于 ITEM1 的值的个数，如果 ITEM1 为 Null，则返回 Null。
count_less_than(ITEM1, LIST)	整数	返回字段列表中小于 ITEM1 的值的个数，如果 ITEM1 为 Null，则返回 Null。
count_not_equal(ITEM1, LIST)	整数	返回字段列表中不等于 ITEM1 的值的个数，如果 ITEM1 为 Null，则返回 Null。
count_nulls(LIST)	整数	返回字段列表中 Null 值的个数。
count_non_nulls(LIST)	整数	返回字段列表中非 Null 值的个数。
date_before(DATE1, DATE2)	布尔逻辑	用于检查日期值的排序。DATE1 早于 DATE2，则返回真值。
first_index(ITEM, LIST)	整数	返回字段列表中包含 ITEM 的第一个字段的索引，如果找不到该值，则返回 0。仅支持字符串、整数和实数类型。 有关详细信息，请参阅第 7 章中的多个响应数据的处理中的 IBM SPSS Modeler 15 用户指南。
first_non_null(LIST)	所有	返回所提供字段列表中的第一个非空值。支持所有存储类型。
first_non_null_index(LIST)	整数	返回字段列表中包含非空值的第一个字段的索引，如果所有值都为空值，则返回 0。支持所有存储类型。
ITEM1 = ITEM2	布尔逻辑	为 ITEM1 等于 ITEM2 的记录返回真值。
ITEM1 /= ITEM2	布尔逻辑	两个字符串不完全相同时返回真值；完全相同时返回 0。
ITEM1 < ITEM2	布尔逻辑	为 ITEM1 小于 ITEM2 的记录返回真值。
ITEM1 <= ITEM2	布尔逻辑	为 ITEM1 小于等于 ITEM2 的记录返回真值。

函数	结果	描述
ITEM1 > ITEM2	布尔逻辑	为 ITEM1 大于 ITEM2 的记录返回真值。
ITEM1 >= ITEM2	布尔逻辑	为 ITEM1 大于等于 ITEM2 的记录返回真值。
last_index(ITEM, LIST)	整数	返回字段列表中包含 ITEM 的最后一个字段的索引，如果找不到该值，则返回 0。仅支持字符串、整数和实数类型。有关详细信息，请参阅第 7 章中的 多个响应数据的处理中的 IBM SPSS Modeler 15 用户指南 。
last_non_null(LIST)	所有	返回所提供字段列表中的最后一个非空值。支持所有存储类型。
last_non_null_index(LIST)	整数	返回字段列表中包含非空值的最后一个字段的索引，如果所有值都为空值，则返回 0。支持所有存储类型。
max(ITEM1, ITEM2)	所有	返回 ITEM1 和 ITEM2 两项中较大的一个。
max_index(LIST)	整数	返回数值字段列表中包含最大值的字段索引，如果所有值为空值，则返回空值。例如，如果所列的第三个字段包含最大值，将返回索引值 3。如果有多个字段包含最大值，则将返回第一个列出的字段（位于最左侧）。有关详细信息，请参阅第 7 章中的 多个响应数据的处理中的 IBM SPSS Modeler 15 用户指南 。
max_n(LIST)	Number	返回数值字段列表中的最大值，如果所有字段值为 Null，则返回 Null。有关详细信息，请参阅第 7 章中的 多个字段的汇总中的 IBM SPSS Modeler 15 用户指南 。
member(ITEM, LIST)	布尔逻辑	如果 ITEM 是指定 LIST 的成员，则返回真值。否则返回假值。还可以指定字段名称列表。有关详细信息，请参阅第 7 章中的 多个字段的汇总中的 IBM SPSS Modeler 15 用户指南 。
min(ITEM1, ITEM2)	所有	返回 ITEM1 和 ITEM2 两项中较小的一个。
min_index(LIST)	整数	返回数值字段列表中包含最小值的字段索引，如果所有值为空值，则返回空值。例如，如果所列的第三个字段包含最小值，将返回索引值 3。如果有多个字段包含最小值，则将返回第一个列出的字段（位于最左侧）。有关详细信息，请参阅第 7 章中的 多个响应数据的处理中的 IBM SPSS Modeler 15 用户指南 。
min_n(LIST)	Number	返回数值字段列表中的最小值，如果所有字段值为 Null，则返回 Null。
time_before(TIME1, TIME2)	布尔逻辑	用于检查时间值的排序。TIME1 早于 TIME2，则返回真值。
value_at(INT, LIST)		在偏移 INT 处返回每个列出字段的值；如果偏移超出了有效值的范围（即小于 1 或大于所列字段的个数），则返回空值。支持所有存储类型。

逻辑函数

CLEM 表达式可用来执行逻辑运算。

函数	结果	描述
COND1 and COND2	布尔逻辑	此运算为逻辑合取运算，当 COND1 和 COND2 同时为真时，返回真值。如果 COND1 为假，则不求 COND2 的值；此时可以构造这样的合取，让 COND1 首先检验 COND2 中运算是合法的。例如， <code>length(Label) >=6</code> 和 <code>Label(6) = 'x'</code> 。
COND1 or COND2	布尔逻辑	此运算为逻辑（包含性）析取运算，当 COND1 或 COND2 为真或这两者同时为真时，返回真值。如果 COND1 为真，则不求 COND2 的值。
not (COND)	布尔逻辑	此运算为逻辑否运算，如果 COND 为假，则返回真。否则，此运算将返回值 0。
if COND then EXPR1 else EXPR2 endif	所有	此运算为条件求值。如果 COND 为真，此运算将返回 EXPR1 的结果。否则，将返回 EXPR2 的求值结果。
if COND1 then EXPR1 elseif COND2 then EXPR2 else EXPR_N endif	所有	此运算为多分支条件求值。如果 COND1 为真，此运算将返回 EXPR1 的结果。否则，如果 COND2 为真，此运算将返回 EXPR2 的求值结果。否则，将返回 EXPR_N 的求值结果。

数值函数

CLEM 包含许多常用数值函数。

函数	结果	描述
- NUM	Number	用于求 NUM 的相反数。返回具有相反符号的对应数字。
NUM1 + NUM2	Number	返回 NUM1 与 NUM2 相加所得的和。
code - NUM2	Number	返回 NUM2 减去 NUM1 所得的差。
NUM1 * NUM2	Number	返回 NUM1 乘以 NUM2 所得的值。
NUM1 / NUM2	Number	返回 NUM1 除以 NUM2 所得的商。
INT1 div INT2	Number	用于进行整数除法。返回 INT1 除以 INT2 所得的商。
INT1 rem INT2	Number	返回 INT1 除以 INT2 所得的余数。例如， <code>INT1 - (INT1 div INT2) * INT2</code> 。
INT1 mod INT2	Number	此函数已被拒绝。改用 <code>rem</code> 函数。
BASE ** POWER	Number	返回 BASE 的 POWER 次方，底和幂可以是任意数字（一种情况除外：如果 POWER 为除整数 0 之外其他类型的 0，则 BASE 不能为 0）。如果 POWER 为整数，则可通过将 BASE 连续乘 POWER 次计算出结果。因此，如果 BASE 为整数，则结果为整数。如果 POWER 为整数 0，则结果将始终是与 BASE 类型相同的 1。如果 POWER 不是整数，则用公式 <code>exp(POWER * log(BASE))</code> 计算结果。
abs (NUM)	Number	返回 NUM 的绝对值（始终为同一类型的数值）。
exp (NUM)	实数	返回 e 的 NUM 次幂，其中 e 是自然对数的底数。
fracof (NUM)	实数	返回 NUM 的小数部分，定义为 <code>NUM - intof (NUM)</code> 。
intof (NUM)	整数	将其参数截为整数。返回与 NUM 符号相同的整数，取满足 <code>abs (INT) <= abs (NUM)</code> 的最大值。

函数	结果	描述
log(NUM)	实数	返回 NUM 的自然对数（以 e 为底），NUM 不能为零（无论何种类型）。
log10(NUM)	实数	返回 NUM 的对数（以 10 为底），不能为零。此函数定义为 $\log(\text{NUM}) / \log(10)$ 。
negate(NUM)	Number	用于求 NUM 的相反数。返回具有相反符号的对应数字。
round(NUM)	整数	用于将 NUM 四舍五入取整数，如果 NUM 为正数，则采用公式 $\text{intof}(\text{NUM}+0.5)$ ；如果 NUM 为负数，则采用公式 $\text{intof}(\text{NUM} - 0.5)$ 。
sign(NUM)	Number	用于确定 NUM 的符号。如果 NUM 为整数，此运算将返回 -1、0 或 1。如果 NUM 为实数，将返回 -1.0、0.0 或 1.0，取决于 NUM 是负数、零还是正数。
sqrt(NUM)	实数	返回 NUM 的平方根。NUM 必须是正数。
sum_n(LIST)	Number	返回数值字段列表中字段值的和，如果所有字段均为 Null，则返回 Null 值。有关详细信息，请参阅第 7 章中的多个字段的汇总中的 IBM SPSS Modeler 15 用户指南。
mean_n(LIST)	Number	返回数值字段列表中字段值的均值，如果所有字段值均为 Null 值，则返回 Null 值。
sdev_n(LIST)	Number	返回数值字段列表中字段值的标准差，如果所有字段值均为 Null 值，则返回 Null 值。

三角函数

本部分介绍的函数以角度为参数或返回结果为角度。无论哪种情况，角度单位（弧度或度数）均由相关流选项的设置控制。

函数	结果	描述
arccos(NUM)	实数	计算指定角度的反余弦值。
arccosh(NUM)	实数	计算指定角度的反双曲余弦值。
arcsin(NUM)	实数	计算指定角度的反正弦值。
arcsinh(NUM)	实数	计算指定角度的反双曲正弦值。
arctan(NUM)	实数	计算指定角度的反正切值。
arctan2(NUM_Y, NUM_X)	实数	计算 $\text{NUM_Y} / \text{NUM_X}$ 的反正切值，然后使用两个数的符号推导出象限信息。结果是范围在 $-\pi < \text{ANGLE} \leq \pi$ (radians) - $180 < \text{ANGLE} \leq 180$ (degrees) 间的实数
arctanh(NUM)	实数	计算指定角度的反双曲正切值。
cos(NUM)	实数	计算指定角度的余弦值。
cosh(NUM)	实数	计算指定角度的双曲余弦值。
pi	实数	此常数是最接近 pi 的实数。
sin(NUM)	实数	计算指定角度的正弦值。
sinh(NUM)	实数	计算指定角度的双曲正弦值。
tan(NUM)	实数	计算指定角度的正切值。
tanh(NUM)	实数	计算指定角度的双曲正切值。

概率函数

概率函数返回基于各种分布的概率，例如，学生氏 t 分布中某个值将会小于特定值的概率。

函数	结果	描述
<code>cdf_chi_sq(NUM, DF)</code>	实数	返回具有指定自由度的卡方分布中某个值将会小于指定数值的概率。
<code>cdf_f(NUM, DF1, DF2)</code>	实数	返回自由度为 $DF1$ 和 $DF2$ 的 F 分布中某个值将会小于指定数值的概率。
<code>cdf_normal(NUM, MEAN, STDDEV)</code>	实数	返回具有指定均值和标准差的正态分布中某个值将会小于指定数值的概率。
<code>cdf_t(NUM, DF)</code>	实数	返回具有指定自由度的学生氏 t 分布中某个值将会小于指定数值的概率。

位元整数运算

借助上述函数，可以按表示二进制补码值的位模式（其中位的位置 N 的权重为 2^{*N} ）来操控整数。位从 0 开始往上数。这些运算就好像是把整数的符号位向左无限延伸。因此，最高有效位之前的所有位，正整数均为 0，负整数均为 1。

注意：不能从脚本中调用位元函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
<code>~~ INT1</code>	整数	产生整数 $INT1$ 的位元补码。即， $INT1$ 的对应位为 0 的结果均为 1。 $~~ INT = -(INT + 1)$ 始终成立。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
<code>INT1 INT2</code>	整数	此运算的结果是 $INT1$ 和 $INT2$ 的位元“或”。即， $INT1$ 和 $INT2$ 其中一个的对应位上是 1 或两者的对应位上同时是 1 时“或”后的结果为 1。
<code>INT1 /& INT2</code>	整数	此运算的结果是 $INT1$ 和 $INT2$ 的位元“异或”。即， $INT1$ 和 $INT2$ 其中一个的对应位是 1 但非同时为 1 时“异或”后的结果为 1。
<code>INT1 && INT2</code>	整数	产生整数 $INT1$ 和 $INT2$ 的位元“与”。即， $INT1$ 和 $INT2$ 的对应位同时为 1 时进行这一运算后的结果才为 1。
<code>INT1 &&~~ INT2</code>	整数	将对 $INT1$ 和 $INT2$ 位元求补后的结果进行位元“与”运算。即， $INT1$ 的对应位是 1 且 $INT2$ 的对应位是 0 时进行这一运算后的结果才为 1。此运算与 $INT1 \&& (~~ INT2)$ 运算的结果相同，可用于清除在 $INT2$ 中设置的 $INT1$ 的位。
<code>INT << N</code>	整数	产生 $INT1$ 向左偏移 N 个位置的位元模式。如果 N 为负值，则向右移。
<code>INT >> N</code>	整数	产生 $INT1$ 向右偏移 N 个位置的位元模式。如果 N 为负值，则向左移。
<code>INT1 &&=_0 INT2</code>	布尔逻辑	相当于布尔表达式 $INT1 \&& INT2 \neq 0$ ，但效率更高。

函数	结果	描述
<code>INT1 &&/=_0 INT2</code>	布尔逻辑	相当于布尔表达式 <code>INT1 && INT2 == 0</code> ，但效率更高。
<code>integer_bitcount(INT)</code>	整数	计算 INT 的二进制补码 表示中数字 1 或 0 的个数。如果 INT 是非负数，则 N 为 1 的个数。如果 INT 为负数，则 N 为 0 的个数。由于符号扩展的原因，非负整数中将有无数个 0，而负整数中将有无数个 1。对于此运算，始终有 <code>integer_bitcount(INT) = integer_bitcount(-(INT+1))</code> 成立。
<code>integer_leastbit(INT)</code>	整数	返回整数 INT 中最低有效位组的数位位置 N。N 是 2 的最大权重，据此 INT 将进行准确划分。
<code>integer_length(INT)</code>	整数	以二进制补码 INT 形式返回整数的数位长度。即，N 是最小整数，以便 <code>INT < (1 << N) if INT >= 0</code> <code>INT >= (-1 << N) if INT < 0</code> 。如果 INT 为非负数，则表示为无符号整数的 INT 需要一个至少 N 位的字段。另外，最少需要用 N+1 个数位来表示有符号整数 INT（不论符号是正是负）。
<code>testbit(INT, N)</code>	布尔逻辑	检验整数 INT 中位置 N 处的数位，并以布尔值的形式（真表示状态为 1，假表示状态为 0）返回数位 N 的状态。

随机函数

以下函数用于随机选择项目或随机生成数值。

函数	结果	描述
<code>oneof(LIST)</code>	所有	返回一个从 LIST 中随机选取的元素。应按 <code>[ITEM1, ITEM2, ..., ITEM_N]</code> 方式输入列表项。注意，还可以指定字段名列表。有关详细信息，请参阅第 7 章中的多个字段的汇总中的 IBM SPSS Modeler 15 用户指南。
<code>random(NUM)</code>	Number	返回类型相同（INT 或 REAL）的均匀分布随机数，其范围从 1 至 NUM。如果使用的是整数，则仅返回整数。如果使用（十进制）实数，则返回实数（由流选项确定的十进制精度）。此函数所返回的最大随机数可以等于 NUM。
<code>random0(NUM)</code>	Number	此函数的属性与 <code>random(NUM)</code> 相同，但将从 0 开始取值。它所返回的最大随机数决不会等于 X。

字符串函数

在 CLEM 中您可以对字符串执行以下操作：

- 比较字符串
- 创建字符串
- 访问字符

在 CLEM 中，字符串是一对英文双引号 ("**string quotes**") 之间的一组字符序列。这里所用的字符 (CHAR) 可以是任何一个字母数字字符。这些字符已在 CLEM 表达式中使用反单引号并以 ``character`` 的形式声明过，例如 ``z``、``A`` 或 ``2``。如果字符超出边界或为字符串中的负指数，则会导致未定义的行为。

注意：在使用与不使用 SQL 回送的字符串之间的对比可能生成存在尾部空格的不同结果。

函数	结果	描述
<code>allbutfirst(N, STRING)</code>	字符串	返回 STRING 删除前 N 个字符的字符值。
<code>allbutlast(N, STRING)</code>	字符串	返回字符串 STRING 除去后 N 个字符后所生成的新字符串。
<code>alphabefore(STRING1, STRING2)</code>	布尔逻辑	用于检查字符串的数字字母顺序。如果 STRING1 在 STRING2 之前，则返回真值。
<code>endstring(LENGTH, STRING)</code>	字符串	从指定字符串中抽取后 N 个字符。如果字符串长度小于或等于指定长度，则字符串不发生改变。
<code>hasendstring(STRING, SUBSTRING)</code>	整数	此函数的功能等同于 <code>isendstring(SUBSTRING, STRING)</code> 。
<code>hasmidstring(STRING, SUBSTRING)</code>	整数	此函数的功能等同于 <code>ismidstring(SUBSTRING, STRING)</code> （嵌入的子字符串）。
<code>hasstartstring(STRING, SUBSTRING)</code>	整数	此函数的功能等同于 <code>isstartstring(SUBSTRING, STRING)</code> 。
<code>hassubstring(STRING, N, SUBSTRING)</code>	整数	此函数的功能等同于 <code>issubstring(SUBSTRING, N, STRING)</code> ，这里 N 的默认值为 1。
<code>count_substring(STRING, SUBSTRING)</code>	整数	返回字符串中指定字符串出现的次数。例如， <code>count_substring("foooo.txt", "oo")</code> 返回 3。
<code>hassubstring(STRING, SUBSTRING)</code>	整数	此函数的功能等同于 <code>issubstring(SUBSTRING, 1, STRING)</code> ，这里 N 的默认值为 1。
<code>isalphacode(CHAR)</code>	布尔逻辑	如果 CHAR 是字符代码为字母的指定字符串（常为字段名）中的某个字符，则返回真值。否则，此函数的返回结果为 0。例如， <code>isalphacode(produce_num(1))</code> 。
<code>isendstring(SUBSTRING, STRING)</code>	整数	如果字符串 STRING 以子字符串 SUBSTRING 结尾，则该函数将返回字符串 STRING 中子字符串 SUBSTRING 的整数下标。否则，此函数的返回结果为 0。
<code>islowercode(CHAR)</code>	布尔逻辑	如果 CHAR 是指定字符串（通常为字段名）中的某个小写字母字符，则将返回真值。否则，该函数的返回值为 0，例如 <code>islowercode(`)`</code> 和 <code>islowercode(country_name(2))</code> 均为有效表达式。
<code>ismidstring(SUBSTRING, STRING)</code>	整数	如果 SUBSTRING 是 STRING 的子字符串，但首字符或尾端字符与 STRING 不同，则该函数将返回此子字符串的起始下标。否则，此函数的返回结果为 0。

函数	结果	描述
isnumbercode (CHAR)	布尔逻辑	如果指定字符串（通常为字段名）的 CHAR 为数字字符，则返回真值。否则，此函数的返回结果为 0。例如， <code>isnumbercode (product_id (2))</code> 。
isstartstring (SUBSTRING, STRING)	整数	如果字符串 STRING 以子字符串 SUBSTRING 开头，则该函数将返回下标 1。否则，此函数的返回结果为 0。
issubstring (SUBSTRING, N, STRING)	整数	从字符串 STRING 中搜索（从第 N 个字符开始）等于 SUBSTRING 的子字符串。如果找到子字符串，则该函数返回匹配子字符串的起始整数下标。否则，此函数将返回值 0。如果未指定 N，此函数的默认值为 1。
issubstring (SUBSTRING, STRING)	整数	从字符串 STRING 中搜索（从第 N 个字符开始）等于 SUBSTRING 的子字符串。如果找到子字符串，则该函数返回匹配子字符串的起始整数下标。否则，此函数将返回值 0。如果未指定 N，此函数的默认值为 1。
issubstring_count (SUBSTRING, N, STRING) :	整数	返回 STRING 中第 N 次出现的 SUBSTRING 的索引。如果 SUBSTRING 的出现次数少于 N 次，则返回结果为 0。
issubstring_lim (SUBSTRING, N, STARTLIM, ENDLIM, STRING)	整数	此函数的功能与 <code>issubstring</code> 相同，但匹配必须从下标 STARTLIM 或其之前开始，并且必须从下标 ENDLIM 或其之前结束。为任一参数提供假值，则可禁用 STARTLIM 或 ENDLIM 限制。例如， <code>issubstring_lim (SUBSTRING, N, false, false, STRING)</code> 函数功能等同于 <code>issubstring</code> 。
isuppercode (CHAR)	布尔逻辑	如果 CHAR 是大写字母字符，则返回真值。否则，该函数的返回值为 0，例如 <code>isuppercode (`)</code> 和 <code>isuppercode (country_name (2))</code> 均为有效表达式。
last (CHAR)	字符串	返回 STRING（长度至少为一个字符）的最后一个字符 CHAR。
length (STRING)	整数	返回字符串 STRING 的长度，即字符串内的字符数。
locchar (CHAR, N, STRING)	整数	用于标识符号字段中的字符位置。此函数将在字符串 STRING 中搜索字符 CHAR（从 STRING 的第 N 个字符开始搜索）。此函数的返回值表示（从 N 开始）找到字符的位置。如果未找到字符，则函数返回结果为 0。如果此函数存在无效偏移 (N)（例如，某个偏移超出了字符串长度），则函数返回结果为 <code>\$null\$</code> 。 例如， <code>locchar (`n`, 2, web_page)</code> 在名为 <code>web_page</code> 的字段中从字段值的第二个字符开始搜索字符 <code>`n`</code> 。 注意：请用反单引号将指定字符引起。

函数	结果	描述
<code>locchar_back (CHAR, N, STRING)</code>	整数	与 <code>locchar</code> 函数类似，所不同的是搜索反向进行并且从第 N 个字符开始。例如， <code>locchar_back('n', 9, web_page)</code> 将从 <code>web_page</code> 的第 9 个字符开始搜索并且是从后向前移动搜索。如果此函数存在无效偏移（例如，某个偏移超出了字符串长度），则函数返回结果为 <code>\$null\$</code> 。理想情况下，您将 <code>locchar_back</code> 与 <code>length(<field>)</code> 函数结合使用，以便动态地使用字段当前值的长度。例如， <code>locchar_back('n', (length(web_page)), web_page)</code> 。
<code>lowertoupper (CHAR)</code> <code>lowertoupper (STRING)</code>	CHAR 或 String	可以输入字符串或字符，此函数将返回一个类型相同项，小写字符将转换为相应的大写字符。例如， <code>lowertoupper('a')</code> 、 <code>lowertoupper("My string")</code> 或 <code>lowertoupper(field_name(2))</code> 均为有效表达式。
<code>matches</code>	布尔逻辑	如果字符串与指定模式相匹配，则将返回真值。模式必须是字符串字符，而不能是包含模式的字段名。问号 (?) 可以包含在模式中以便匹配某一字符，星号 (*) 可以匹配零个或多个字符。要匹配文字问号或星号（而不是将其用作通配符），可以使用反斜杠 (\) 作为转义符号。
<code>replace (SUBSTRING, NEWSUBSTRING, STRING)</code>	字符串	在指定 STRING 中，用 NEWSUBSTRING 替换 SUBSTRING 中的所有示例。
<code>replicate (COUNT, STRING)</code>	字符串	返回一个由原始字符串指定个数的复本所构成的字符串。
<code>stripchar (CHAR, STRING)</code>	字符串	借助此函数，您可从某个字符串或字段中除去指定字符。例如，您可以使用此函数从数据中除去多余符号（如货币符号）以获取一个简单的数值或名称。例如，通过语法 <code>stripchar('\$', 'Cost')</code> 可以除去所有值中的美元符号并返回一个新字段。 注意：请用反单引号将指定字符引起。
<code>skipchar (CHAR, N, STRING)</code>	整数	在字符串 STRING 中搜索除 CHAR 以外的字符（从第 N 个字符开始搜索）。此函数返回一个表示发现子字符串位置的整数子字符串，如果从第 N 个位置开始的每个字符是 CHAR，则返回 0。如果此函数存在无效偏移（例如，某个偏移超出了字符串长度），则函数返回结果为 <code>\$null\$</code> 。 <code>locchar</code> 函数经常与 <code>skipchar</code> 结合使用以确定 N（开始搜索字符串的位置）的值。例如， <code>skipchar('s', (locchar('s', 1, "MyString")))</code> 、 <code>"MyString"</code> 。
<code>skipchar_back (CHAR, N, STRING)</code>	整数	与 <code>skipchar</code> 函数类似，所不同的是搜索反向进行并且从第 N 个字符开始。

函数	结果	描述
startstring(LENGTH, STRING)	字符串	从指定字符串中抽取前 N 个字符。如果字符串长度小于或等于指定长度，则字符串不发生改变。
strmember (CHAR, STRING)	整数	相当于 locchar (CHAR, 1, STRING)。此函数返回整数子字符串或 0，返回的整数子字符串表示 CHAR 第一次出现的位置。如果函数存在无效偏移（例如，某个偏移超出了字符串长度），则函数返回结果为 \$null\$。
subscrs (N, STRING)	CHAR	返回输入字符串 STRING 的第 N 个字符 CHAR。此函数也可采用简写形式 STRING (N)。例如，lowertoupper (“name” (1)) 是一个有效的变量名。
substring (N, LEN, STRING)	字符串	返回从位于下标 N 的字符开始的字符串 SUBSTRING，此子字符串由字符串 STRING 的 LEN 个字符组成。
substring_between (N1, N2, STRING)	字符串	返回 STRING 的子字符串，该字符串从下标 N1 开始并结束于下标 N2。
trim (STRING)	字符串	除去指定字符串的前导和尾部空格。
trim_start (STRING)	字符串	除去指定字符串的前导空格。
trimend (STRING)	字符串	除去指定字符串的尾部空格。
unicode_char (NUM)	CHAR	返回 Unicode 值为 NUM 的字符。
unicode_value (CHAR)	NUM	返回 CHAR 的 Unicode 值
uppertolower (CHAR) uppertolower (STRING)	CHAR 或 String	可以输入字符串或字符，此函数将返回一个类型相同项，大写字符将转换为相应的小写字符。 注意：指定字符串时一定要用英文双引号，而指定字符时一定要用反向单引号。简单字段名不必使用引号。

SoundEx 函数

SoundEx 是在知道字符串的发音却不知其准确拼写的情况下用来查找字符串的一种方法。此种方法在 1918 年提出，它在对特定字母发音进行语音假定的基础上搜索出类似发音的字。此种方法可用于在数据库中搜索名称，例如相似名称的拼写和发音可能有所不同。尽管存在限制（例如，即便前导字母组合 ph 和 f 的发音相同，但它们仍然不匹配），基本的 SoundEx 算法在多个源中均有归档，并且绝大多数数据库均以某种形式支持此种算法。

函数	结果	描述
soundex (STRING)	整数	返回指定 STRING 的四字符 SoundEx 代码。
soundex_difference (STRING1, STRING2)	整数	返回 0 和 4 之间的某个整数，此整数表示两个字符串中 SoundEx 编码相同的字符的个数，0 表示不相似，4 表示两个字符串极相似或完全相同。

日期和时间函数

CLEM 包含一系列处理含有字符串日期时间存储变量的字段的函数，这些字符串变量代表日期和时间。可在“流属性”对话框中指定具体到每个流的日期和时间格式。日期和时间函数根据当前选定的格式来解析日期和时间字符串。

如果用两位数指定日期中的年（即未指定世纪），则 IBM® SPSS® Modeler 将采用“流属性”对话框中所指定的默认世纪。

注意：不能从脚本中调用日期和时间函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
@TODAY	字符串	如果在“流属性”对话框中选择翻转日/分钟，该函数将返回采用当前日期格式的当前日期字符串。如果您采用两位数日期格式且未选择翻转日/分钟，该函数将在当前服务器上返回 \$null\$。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
to_time (ITEM)	时间	将指定字段的存储值转换为时间。
to_date (ITEM)	日期	将指定字段的存储值转换为日期。
to_timestamp (ITEM)	时间戳	将指定字段的存储值转换为时间戳。
to_datetime (ITEM)	日期时间	将指定字段的存储值转换为日期、时间或时间戳值。
datetime_date (ITEM)	日期	返回数字、字符串或时间戳的日期值。请注意，这是唯一允许您将数字（以秒数记）转换回日期的函数。如果 ITEM 为字符串，将以当前日期格式解析字符串来创建日期。只有“流属性”对话框中指定的日期格式正确，此函数方能成功执行。如果 ITEM 为数字，将解析为自基线日期（或纪元）以来的秒数。不足一天的部分将被截断。如果 ITEM 为时间戳，将返回时间戳的日期部分。如果 ITEM 为日期，将返回未更改的日期。
date_before (DATE1, DATE2)	布尔逻辑	如果 DATE1 所表示的日期或时间戳在 DATE2 之前，则返回真值。否则，此函数的返回结果为 0。
date_days_difference (DATE1, DATE2)	整数	以整数的形式返回从日期或时间戳 DATE1 到日期或时间戳 DATE2 的天数。如果 DATE2 在 DATE1 之前，则该函数返回负值。
date_in_days (DATE)	整数	以整数形式返回从基线日期到日期或时间戳 DATE 的天数。如果 DATE 在基线日期之前，则该函数返回负数。为了计算正常进行，必须包含有效日期。例如，不能将 2001 年 2 月 29 日指定为日期。因为 2001 年并非闰年，所以上述日期不存在。
date_in_months (DATE)	实数	以实数形式返回从基线日期到日期或时间戳 DATE 的月数。这是基于每月 30.4375 天的近似数字。如果 DATE 在基线日期之前，则该函数返回负数。为了计算正常进行，必须包含有效日期。例如，不能将 2001 年 2 月 29 日指定为日期。因为 2001 年并非闰年，所以上述日期不存在。

函数	结果	描述
date_in_weeks (DATE)	实数	以实数形式返回从基线日期到日期或时间戳 DATE 的周数。这基于每周 7.0 天。如果 DATE 在基线日期之前，则该函数返回负数。为了计算正常进行，必须包含有效日期。例如，不能将 2001 年 2 月 29 日指定为日期。因为 2001 年并非闰年，所以上述日期不存在。
date_in_years (DATE)	实数	以实数形式返回从基线日期到日期或时间戳 DATE 的年数。这是基于每年 365.25 天的近似数字。如果 DATE 在基线日期之前，则该函数返回负数。为了计算正常进行，必须包含有效日期。例如，不能将 2001 年 2 月 29 日指定为日期。因为 2001 年并非闰年，所以上述日期不存在。
date_months_difference (DATE1, DATE2)	实数	以实数形式返回从日期或时间戳 DATE1 到日期或时间戳 DATE2 的月数。这是基于每月 30.4375 天的近似数字。如果 DATE2 在 DATE1 之前，则该函数返回负值。
datetime_date (YEAR, MONTH, DAY)	日期	创建给定 YEAR、MONTH、DAY 的日期值。参数必须为整数。
datetime_day (DATE)	整数	对给定的 DATE 或时间戳，返回其所在月份的天数。返回结果是范围为 1 至 31 的整数。
datetime_day_name (DAY)	字符串	返回给定 DAY 的全称。参数必须为范围 1（周日）到 7（周六）之间的整数。
datetime_hour (TIME)	整数	返回某个 TIME 或时间戳的小时部分。返回结果是范围为 0 至 23 的整数。
datetime_in_seconds (TIME)	实数	返回 TIME 存储的秒钟部分。
datetime_in_seconds (DATE), datetime_in_seconds (DATE-TIME)	实数	返回从当前 DATE 或 DATETIME 到基线日期 (1900-01-01) 之间的差值的累积秒数。
datetime_minute (TIME)	整数	返回某个 TIME 或时间戳的分钟部分。返回结果是范围为 0 至 59 的整数。
datetime_month (DATE)	整数	返回某个 DATE 或时间戳的月份部分。返回结果是范围为 1 至 12 的整数。
datetime_month_name (MONTH)	字符串	返回给定 MONTH 的全称。参数必须为范围 1 到 12 内的整数。
datetime_now	时间戳	以时间戳形式返回当前时间。
datetime_second (TIME)	整数	返回某个 TIME 或时间戳的秒钟部分。返回结果是范围为 0 至 59 的整数。
datetime_day_short_name (DAY)	字符串	返回给定 DAY 的简称。参数必须为范围 1（周日）到 7（周六）之间的整数。
datetime_month_short_name (MONTH)	字符串	返回给定 MONTH 的简称。参数必须为范围 1 到 12 内的整数。
datetime_time (HOUR, MINUTE, SECOND)	时间	返回给定 HOUR、MINUTE 和 SECOND 的时间值。参数必须为整数。
datetime_time (ITEM)	时间	返回给定 ITEM 的时间值。
datetime_timestamp (YEAR, MONTH, DAY, HOUR, MINUTE, SECOND)	时间戳	返回指定 YEAR、MONTH、DAY、HOUR、MINUTE 以及 SECOND 的时间戳值。
datetime_timestamp (DATE, TIME)	时间戳	返回给定 DATE 和 TIME 的时间戳值。

函数	结果	描述
<code>datetime_timestamp</code> (NUMBER)	时间戳	返回给定秒数 NUMBER 的时间戳值。
<code>datetime_weekday</code> (DATE)	整数	对给定 DATE 或时间戳，返回其所在周的天数数据。
<code>datetime_year</code> (DATE)	整数	返回某个 DATE 或时间戳的年份部分。结果为一个整数，如 2002。
<code>date_weeks_difference</code> (DATE1, DATE2)	实数	以实数形式返回从日期或时间戳 DATE1 到日期或时间戳 DATE2 的周数。这基于每周 7.0 天。如果 DATE2 在 DATE1 之前，则该函数返回负值。
<code>date_years_difference</code> (DATE1, DATE2)	实数	以实数形式返回从日期或时间戳 DATE1 到日期或时间戳 DATE2 的年数。这是基于每年 365.25 天的近似数字。如果 DATE2 在 DATE1 之前，则该函数返回负值。
<code>time_before</code> (TIME1, TIME2)	布尔逻辑	如果 TIME1 所表示的时间或时间戳在 TIME2 之前，则返回真值。否则，此函数的返回结果为 0。
<code>time_hours_difference</code> (TIME1, TIME2)	实数	以实数的形式返回时间或时间戳 TIME1 和 TIME2 之间的小时差。如果在“流属性”对话框中选择翻转日/分钟，则用 TIME1 的较大值表示前一天。如果不选择翻转日选项，则较大的 TIME1 值将造成返回值为负数。
<code>time_in_hours</code> (TIME)	实数	以实数的形式返回 TIME 所表示的小时数。例如，采用时间格式 HHMM 时，表达式 <code>time_in_hours('0130')</code> 的返回值为 1.5。TIME 可表示时间或时间戳。
<code>time_in_mins</code> (TIME)	实数	以实数的形式返回 TIME 所表示的分钟数。TIME 可表示时间或时间戳。
<code>time_in_secs</code> (TIME)	整数	以实数的形式返回 TIME 所表示的秒数。TIME 可表示时间或时间戳。
<code>time_mins_difference</code> (TIME1, TIME2)	实数	以实数的形式返回时间或时间戳 TIME1 和 TIME2 之间的分钟差。如果在“流选项”对话框中选择翻转日/分钟，则将采用 TIME1 的较大值来指代前一天（仅当在当前格式中指定分和秒时，指代前一小时）。如果不选择翻转日选项，TIME1 的较大值会造成返回值为负数。
<code>time_secs_difference</code> (TIME1, TIME2)	整数	以整数的形式返回时间或时间戳 TIME1 和 TIME2 之间的秒数差。如果在“流选项”对话框中选择翻转日/分钟，则将采用 TIME1 的较大值来指代前一天（仅当在当前格式中指定分和秒时，指代前一小时）。如果不选择翻转日选项，则较大的 TIME1 值将造成返回值为负数。

转换日期和时间值

请注意，转换函数（及要求特定类型输入的其他函数，如日期或时间值）取决于“流选项”对话框中指定的当前格式。例如，如果某个名为 DATE 的字段是以 Jan 2003 和 Feb 2003 等值的字符串进行存储的，则可以使用以下函数将其存储为日期格式：

`to_date`(DATE)

为了使此转换生效，请选择匹配的日期格式 `MON YYYY` 作为流的默认日期格式。有关详细信息，请参阅第 5 章中的设置流的常规选项中的 IBM SPSS Modeler 15 用户指南。

例如，若要使用填充节点将字符串值转换为日期，请参阅流 `broadband_create_models.str`，它安装在 `streams` 子文件夹下的 `\Demos` 文件夹中。有关详细信息，请参阅第 14 章中的使用时间序列节点进行预测中的 IBM SPSS Modeler 15 应用程序指南。

存储为数字形式的日期。 请注意，前述示例中的 `DATE` 是字段的名称，而 `to_date` 是 CLEM 函数。如果已将日期存储为数字，则可以使用 `datetime_date` 函数进行转换，其中数字解析为自基线日期（或纪元）以来的秒数。

`datetime_date (DATE)`

通过将日期转换为秒数（再反向转换），您可以进行用当前日期加减固定天数的计算，例如：

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

序列函数

对于某些操作而言，事件序列很重要。通过该应用程序，您可以处理以下记录序列：

- 序列和时间序列
- 序列函数
- 记录索引
- 求值的平均数、总和以及对值进行比较
- 监视变更—区别
- @SINCE
- 偏移值
- 其它序列工具

对于许多应用程序，每个通过流传递的记录可以看成是独立于所有其它记录的单个案例。此种情况下，记录的顺序通常并不重要。

然而对于某些类别的问题，记录序列非常重要。这些一般是发生在时间序列中的情况，其中记录序列代表事件的有序排列。每条记录代表着时间中的某个特定时刻的快照，大部分最重要的信息可能并未包含在瞬时值中，而是包含在随时间的流逝不断变更发展的方式中。

当然，相关参数可能并不是时间。例如，记录可以代表沿着某条线的不同位置对事物所采取的分析，但也将应用相同的原则。

通过以下特征可以立即识别出序列和特殊函数：

- 这些函数的前缀均为 @。
- 函数名称采用大写。

序列函数可以引用节点当前处理的记录、已通过节点的记录、甚至是（某种情况下）尚未通过节点的记录。序列函数可与其他 CLEM 表达式部分自由组合，虽然某些函数对参数有所限制。

示例

您会发现了解自某一特定事件发生或条件为真以来的时间长度非常有用。使用函数 `@SINCE` 可实现上述目的，例如：

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

该函数返回最后一条满足此条件的记录的偏移量，即在此记录之前满足条件的记录数。如果条件从不为真，则 `@SINCE` 返回 `@INDEX + 1`。

有时您可能需要在 `@SINCE` 所用的表达式中引用当前记录的值。此时可以使用函数 `@THIS`，它指定了一个始终应用于当前记录的字段名。要找出最后一条满足 `Concentration` 字段值是当前记录两倍的记录的偏移量，您可使用以下表达式：

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

在有些情况下，当前记录的 `@SINCE` 函数的定义条件为真，例如：

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

基于这种原因，`@SINCE` 函数将不对当前记录的条件求值。如果要对当前记录以及前面记录的条件求值，请使用类似函数 `@SINCE0`；如果当前记录的条件为真，则 `@SINCE0` 返回值为 0。

注意：不得从脚本调用 `@` 函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
<code>MEAN(FIELD)</code>	实数	返回指定字段 <code>FIELD</code> 或 <code>FIELDS</code> 的值的均值。
<code>@MEAN(FIELD, EXPR)</code>	实数	返回当前节点所接收的最后 <code>EXPR</code> 个记录（包括当前记录）的 <code>FIELD</code> 值的均值。 <code>FIELD</code> 必须是数值型字段的名称。 <code>EXPR</code> 可以是任何值为大于 0 的整数的表达式。如果 <code>EXPR</code> 被忽略或超出了目前已接收的记录数，则返回目前为止接收到的所有记录数的平均值。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
<code>@MEAN(FIELD, EXPR, INT)</code>	实数	返回当前节点所接收的最后 <code>EXPR</code> 个记录（包括当前记录）的 <code>FIELD</code> 值的均值。 <code>FIELD</code> 必须是数值型字段的名称。 <code>EXPR</code> 可以是任何值为大于 0 的整数的表达式。如果 <code>EXPR</code> 被忽略或超出了目前已接收的记录数，则返回目前为止接收到的所有记录数的平均值。 <code>INT</code> 指定要向后查看的值的最大数量。这比使用两个参数更为有效。
<code>@DIFF1(FIELD)</code>	实数	返回 <code>FIELD1</code> 的一阶微分差。因此单参数形式将返回该字段当前值和前一个值的差。如果之前的相关记录不存在，则返回 0。
<code>@DIFF1(FIELD1, FIELD2)</code>	实数	两参数形式给出了 <code>FIELD1</code> 对 <code>FIELD2</code> 的一阶微分差。如果之前的相关记录不存在，则返回 0。
<code>@DIFF2(FIELD)</code>	实数	返回 <code>FIELD1</code> 的二阶微分差。因此单参数形式将返回该字段当前值和前一个值的差。如果之前的相关记录不存在，则返回 0。
<code>@DIFF2(FIELD1, FIELD2)</code>	实数	两参数形式给出了 <code>FIELD1</code> 对 <code>FIELD2</code> 的一阶微分差。如果之前的相关记录不存在，则返回 0。

函数	结果	描述
@INDEX	整数	返回当前记录的索引。记录 到达当前节点时，即将索引分配给相应的记录。为第一条记录分配索引 1，每条后续记录的索引以 1 递增。
@LAST_NON_BLANK (FIELD)	所有	根据上游源或类型节点中的定义，返回为非空值的 FIELD 的前一个值。如果目前所读取的 FIELD 中没有非空值，则返回 \$null\$。注意，可以为每个字段单独定义空值（又称用户缺失值）。
@MAX (FIELD)	Number	返回指定 FIELD 的最大值。
@MAX (FIELD, EXPR)	Number	返回目前为止所接收的最后 EXPR 条记录的 FIELD 字段（包括当前字段）的最大值。FIELD 必须是数值型字段的名称。EXPR 可以是值为大于 0 的整数的任意表达式。
@MAX (FIELD, EXPR, INT)	Number	返回目前为止所接收的最后 EXPR 条记录的 FIELD 字段（包括当前字段）的最大值。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前已接收的记录数，则返回目前为止接收到的所有记录数的最大值。INT 指定要向后查看的值的最大数量。这比使用两个参数更为有效。
@MIN (FIELD)	Number	返回指定 FIELD 的最小值。
@MIN (FIELD, EXPR)	Number	返回目前为止所接收的最后 EXPR 条记录的 FIELD 字段（包括当前字段）的最小值。FIELD 必须是数值型字段的名称。EXPR 可以是值为大于 0 的整数的任意表达式。
@MIN (FIELD, EXPR, INT)	Number	返回目前为止所接收的最后 EXPR 条记录的 FIELD 字段（包括当前字段）的最小值。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前已接收的记录数，则返回目前为止接收到的所有记录数的最小值。INT 指定要向后查看的值的最大数量。这比使用两个参数更为有效。
@OFFSET (FIELD, EXPR)	所有	返回当前记录中值为 EXPR 的记录偏移量中的 FIELD 值。正偏移指某条记录已经传递，而负偏移则为某条尚未到达的记录指定了“期待”。例如，@OFFSET (Status, 1) 返回前一条记录中 Status 字段的值，而 @OFFSET (Status, -4) “超前查看”序列的四条记录（即未通过节点传递的记录），以获取字段值。请注意，必须将负（超前查看）偏移指定为常数。仅对于正偏移，EXPR 还可以是任意的 CLEM 表达式，将通过此表达式给出 用于当前记录的偏移值。在此情况下，采用该函数的三参数版本可以提高性能（参阅下个函数）。如果表达式返回的不是非负整数，将产生错误 - 即，计算的期待偏移量不合法。 注意：自引用的 @OFFSET 函数无法使用文字前瞻。例如，在填充节点中，您不能使用诸如 @OFFSET (field1, -2) 等表达式来替换 field1 的值。

函数	结果	描述
@OFFSET (FIELD, EXPR, INT)	所有	与 @OFFSET 函数执行相同的操作，但附带第三个参数 INT（用于指定向后查看值的最大数量）。当从表达式计算偏移时，采用第三个参数可以提高性能。 例如，通过表达式 @OFFSET (Foo, Month, 12)，系统将知道仅保留 Foo 的最后 12 个值，否则，系统将存储所有满足条件的值。如果偏移量为常数—包括必须为常数的负“期待”偏移量—第三个参数将毫无意义，此时应当使用该函数的双参数版本。另请参阅在前述双参数版本中自引用函数的说明。
@SDEV (FIELD)	实数	返回 指定的 FIELD 或 FIELDS 值的标准差。
@SDEV (FIELD, EXPR)	实数	返回当前节点所接收的最后 EXPR 个记录（包括当前记录）的 FIELD 值的标准差。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前所接收的记录数，则返回目前为止接收到的所有记录数的标准差。
@SDEV (FIELD, EXPR, INT)	实数	返回当前节点所接收的最后 EXPR 个记录（包括当前记录）的 FIELD 值的标准差。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前所接收的记录数，则返回目前为止接收到的所有记录数的标准差。INT 指定要向后查看的值的最大数量。这比使用两个参数更为有效。
@SINCE (EXPR)	所有	如果 EXPR（可以是任何的 CLEM 表达式）的结果为真，则返回已传递的记录数。
@SINCE (EXPR, INT)	所有	添加第二个参数（INT）时，用于指定向后查看的最大记录数。如果 EXPR 结果不为真，则 INT 为 @INDEX+1。
@SINCE0 (EXPR)	所有	对于当前记录，@SINCE 不返回任何值；而如果当前记录的 EXPR 结果为真，则 @SINCE0 返回 0。
@SINCE0 (EXPR, INT)	所有	添加第二个参数（INT）时，用于指定向后查看的最大记录数。
@SUM (FIELD)	Number	返回指定字段 FIELD 或 FIELDS 的值的和。
@SUM (FIELD, EXPR)	Number	返回当前节点所接收的最后 EXPR 个记录（包括当前记录）的 FIELD 值的和。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前所接收的记录数，则返回目前为止接收到的所有记录数的和。
@SUM (FIELD, EXPR, INT)	Number	返回当前节点所接收的最后 EXPR 个记录（包括当前记录）的 FIELD 值的和。FIELD 必须是数值型字段的名称。EXPR 可以是任何值为大于 0 的整数的表达式。如果 EXPR 被忽略或超出了目前所接收的记录数，则返回目前为止接收到的所有记录数的和。INT 指定要向后查看的值的最大数量。这比使用两个参数更为有效。
@THIS (FIELD)	所有	返回当前记录中名为 FIELD 的字段的价值。仅用于 @SINCE 表达式。

全局函数

函数 @MEAN、@SUM、@MIN、@MAX 和 @SDEV 多数时候用于处理所有已读取的记录（包括当前记录）。但在某些时候，它可用于检测如何将当前记录中的值与整个数据集中的值进行比较。如果使用设置全局量节点来生成整个数据集中的值，则您可在使用全局函数的 CLEM 表达式中访问这些值。

例如，

@GLOBAL_MAX(Age)

返回数据集中的最大 Age 值，同时表达式

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

将给出该记录 Value 和作为标准偏差的全局平均数之间的差。仅当设置全局量节点计算出全局值后，您方可使用它们。通过单击“流属性”对话框“全局量”选项卡中的 Clear Global Values 按钮，您可以撤消当前所有全局值。

注意：不得从脚本调用 @ 函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
@GLOBAL_MAX(FIELD)	Number	返回超出整个数据集(先前由“设置全局量”节点生成)的 FIELD 的最大值。FIELD 必须是数值型字段的名称。如果未设置相应的全局值，则将出现错误。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
@GLOBAL_MIN(FIELD)	Number	返回超出整个数据集(先前由“设置全局量”节点生成)的 FIELD 的最小值。FIELD 必须是数值型字段的名称。如果未设置相应的全局值，则将出现错误。
@GLOBAL_SDEV(FIELD)	Number	返回整个数据集(先前由设置全局量节点生成)的 FIELD 值的标准差。FIELD 必须是数值型字段的名称。如果未设置相应的全局值，则将出现错误。
@GLOBAL_MEAN(FIELD)	Number	返回超出整个数据集(先前由“设置全局量”节点生成)的 FIELD 的平均值。FIELD 必须是数值型字段的名称。如果未设置相应的全局值，则将出现错误。
@GLOBAL_SUM(FIELD)	Number	返回超出整个数据集(由“设置全局量”节点生成)的 FIELD 值的和。FIELD 必须是数值型字段的名称。如果未设置相应的全局值，则将出现错误。

处理空值和 Null 值的函数

借助 CLEM，您可以指定在某个字段中用作“空值”或缺失值的特定值。下列函数用于处理空值。

注意：不得从脚本调用 @ 函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
@BLANK (FIELD)	布尔逻辑	根据上游类型节点或源节点（“类型”选项卡）中所设置的空值处理规则，对所有空值的记录返回真值。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
@LAST_NON_BLANK (FIELD)	所有	根据上游源或类型节点中的定义，返回为非空值的 FIELD 的前一个值。如果目前所读取的 FIELD 中没有非空值，则返回 \$null\$。注意，可以为每个字段单独定义空值（又称用户缺失值）。
@NULL (FIELD)	布尔逻辑	如果 FIELD 的值为系统缺失值 \$null\$，则返回真值。如果字段值为其他值（包括用户定义空值），则返回假值。如果想同时检查空值和 Null 值，则请使用 @BLANK (FIELD) 和 @NULL (FIELD)。
undef	所有	在 CLEM 中一般用于输入 \$null\$ 值 一例如，在过滤节点中用 Null 值替代空值。

在过滤节点中可以“填写”空值字段。在填充节点和导出节点（仅多种模式）中，特殊的 CLEM 函数 @FIELD 将指向正在检查的当前字段。

特殊字段

特殊函数用于指示所检查的具体字段，或用于生成输入字段列表。例如，当一次导出多个字段时，应使用 @FIELD 函数来指示“对选定字段执行导出操作”。使用表达式 log (@FIELD) 来为每个选定字段导出一个新的日志字段。

注意：不得从脚本调用 @ 函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。

函数	结果	描述
@FIELD	所有	对表达式中指定的所有字段执行操作。注意，不能通过脚本调用此函数。有关详细信息，请参阅第 25 页码第 3 章中的 CLEM 脚本中的表达式。
@TARGET	所有	当 CLEM 表达式用于用户定义的分析函数时，@TARGET 代表目标字段或正在分析的目标/预测对的“正确值”。此函数一般用于分析节点。
@PREDICTED	所有	当 CLEM 表达式用于用户定义的分析函数时，@PREDICTED 代表正在分析的目标/预测对的预测值。此函数一般用于分析节点。
@PARTITION_FIELD	所有	替换当前分区字段的名称。
@TRAINING_PARTITION	所有	返回当前训练分区的值。例如，要用选择节点选择训练记录，请使用 CLEM 表达式： @PARTITION_FIELD = @TRAINING_PARTITION 此表达式将确保无论数据中的每个分区使用哪个值表示，选择节点都将照常运行。

函数	结果	描述
@TESTING_PARTITION	所有	返回当前测试分区的值。
@VALIDATION_PARTITION	所有	返回当前验证分区的值。
@FIELDS_BETWEEN(start, end)	所有	根据数据中字段的自然（即插入）顺序，返回指定开始和结束字段（包含）之间的字段名列表。有关详细信息，请参阅第 7 章中的多个字段的汇总中的 IBM SPSS Modeler 15 用户指南。
@FIELDS_MATCHING(pattern)	所有	返回与指定模式相匹配的字段名的列表。问号 (?) 可以包含在模式中以便匹配某一字符，星号 (*) 可以匹配零个或多个字符。要匹配文字问号或星号（而不是将其用作通配符），可以使用反斜杠 (\) 作为转义符号。有关详细信息，请参阅第 7 章中的多个字段的汇总中的 IBM SPSS Modeler 15 用户指南。
@MULTI_RESPONSE_SET	所有	返回已命名的多响应集的字列表。有关详细信息，请参阅第 7 章中的多个响应数据的处理中的 IBM SPSS Modeler 15 用户指南。

部分 II: 属性参考信息

属性参考信息

属性参考信息概述

可以为节点、流、超节点和工程指定多个不同的属性。某些属性在所有节点中通用，例如“名称”、“注释”和“工具提示”，有些属性则只针对某些特定的节点类型。其它属性涉及高级流操作，例如高速缓存或“超节点”行为。可通过标准用户界面（例如当打开对话框编辑节点选项时）访问属性，还可以多种其它方式使用属性。

- 可通过脚本修改属性（如本章所述）。有关详细信息，请参阅下面的 属性语法。
- 可在“超节点”参数中应用节点属性。有关详细信息，请参阅第 9 章中的使用超节点参数访问节点属性中的 IBM SPSS Modeler 15 源、过程和输出节点。
- 启动 IBM® SPSS® Modeler 时，节点属性还可用作命令行选项（使用 -P 标记）的一部分。

在 SPSS Modeler 的脚本编写环境中，节点和流属性通常称为**通道参数**。在本指南中，它们指的是节点或流的属性。

有关脚本编写语言的更多信息，请参阅第 3 章。

属性语法

属性必须采用如下语法结构：

```
NAME:TYPE.PROPERTY
```

其中，**NAME** 是节点名称，**TYPE** 则是节点类型（例如，`multiplotnode` 或 `derivenode`）。可以省略 **NAME** 或 **TYPE**，但必须至少包含其中一项。**PROPERTY** 指表达式所涉及的节点或流的参数名称。例如，以下语法用于从下游数据中过滤年龄字段：

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

要对某个参数（**NAME**、**TYPE** 或 **PROPERTY**）使用自定义值，首先应设置语句值，例如 `set derive.new_name = mynewfield`。在此之后，您可以将值（`mynewfield`）用作参数，方法是将其置于 ^ 标记之后。例如，可以使用如下语法设置上述名称的导出节点类型：

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

可在参数 **TYPE** 中指定 IBM® SPSS® Modeler 中使用的所有节点，语法如下：
NAME:TYPE.PROPERTY。

结构化属性

脚本编写通过结构化属性，增强语法解析清晰度的方式有二：

- 指定复杂节点属性名称的结构，例如类型节点、过滤节点或平衡节点。
- 提供一种可一次指定多种属性的格式。

复杂接口的结构化

带有表和其它复杂接口的节点（类型、过滤和平衡节点）的脚本必须遵循特定的结构以便正确解析。这些结构化属性的名称要比单个 ID 名称复杂。例如，过滤节点内，每个可用字段（在其上游）均处于开或关的状态。为了引用此信息，过滤节点为每个字段（无论字段真或假）存储一个信息项，由此构成的多个项将由称为**字段**的单个属性来访问和更新。该属性的值可能为（或指定为）`true` 或 `false`。假如过滤节点 `mynode`（在其上游）有一名为年龄的字段。要关闭此字段，请将属性 `mynode.include.Age` 的值设置为 `false`，如下所示：

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

为设置多重属性而结构化

对于多数节点而言，您可以一次分配节点或流的多个属性。这称为**多重设置命令**或**设置块**。有关详细信息，请参阅第 30 页码第 4 章中的 `set` 命令。

某些情况下，结构化属性相当复杂。可以将反斜杠“\”用作行连字符来清晰地排列参数。示例如下：

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

结构化属性的另一个优势在于，在某个节点稳定之前可以在该节点上设置若干个属性。默认情况下，多重设置将在基于单个属性设置的操作运行之前，在块中设置所有属性。例如，如果在定义固定文件节点时分两步设置字段属性，由于节点在两个设置均生效之前是不一致的，将会导致发生错误。以多重设置方式定义属性，可使得在更新数据模型前设置上述两个属性，从而避免上述问题发生。

缩写

在节点属性语法中使用标准缩写。了解缩写有助于构建脚本。

缩写	含义
abs	绝对值
len	长度
min	最小值
max	最大值
correl	Correlation
covar	Covariance
num	数字或数值
pct	百分比
transp	透明度
xval	交叉验证
变量	方差或变量（源节点中）

节点和流属性示例

在 IBM® SPSS® Modeler 中可以各种方式使用节点和流属性。此类属性经常用作脚本的一部分，作为**独立脚本**的一部分用以实现多个流或操作的自动化；或用作**流脚本**的一部分用以实现单个流内部的过程自动化。还可通过在“超节点”内使用节点参数来指定节点参数。就最基础的水平而言，属性还可用作命令行选项来启动 SPSS Modeler。将 `-p` 参数用作命令行调用的一部分时，可以使用流属性来更改流设置。

<code>s.max_size</code>	涉及到节点 <code>s</code> 的属性 <code>max_size</code> 。
<code>s:samplenode.max_size</code>	涉及到节点 <code>s</code> 的属性 <code>max_size</code> ，其必须为样本节点。
<code>:samplenode.max_size</code>	涉及到当前流中抽样节点的属性 <code>max_size</code> （只能有一个样本流）。
<code>s:sample.max_size</code>	涉及到节点 <code>s</code> 的属性 <code>max_size</code> ，其必须为样本节点。
<code>t.direction.Age</code>	涉及到“类型”节点 <code>t</code> 中年龄字段的角色。
<code>:.max_size</code>	*** 非法操作 *** 必须指定节点名或节点类型。

示例 `s:sample.max_size` 说明不一定要写出节点类型的全称。

示例 `t.direction.Age` 说明，当某个节点属性比带有单个值的单个通道复杂时，某些通道名称将自行结构化。此类通道称为**结构化**或**复杂**属性。

节点属性概述

每种类型的节点均有其合法属性集，每种属性也有其类型。该类型可以是常用类型—数字、标志或字符串—此时属性设置将强制至正确类型。如果无法强制操作，则将出现错误。另外，通过属性引用可以指定合法值的范围，例如 `Discard`、`PairAndDiscard` 和 `IncludeAsText`，此时如果采用其它值，则将出现错误。应通过采用值 `true` 或 `false` 来读取或设置标志属性。（设置值时也可识别如下变异值：`Off`、`OFF`、`off`、`No`、`NO`、`no`、`n`、`N`、`f`、`F`、`false`、`False`、`FALSE` 或 `0`，但在某些情况下读取属性值时会出错。所有其他值均视为 `true`。使用 `true` 和 `false` 时保持一致将可以避免混淆。）在本指南的参考表中，属性说明列对结构化属性进行了说明，并给出了属性的使用格式。

通用节点属性

IBM® SPSS® Modeler 中的很多属性通用于所有节点（包括超节点）。

属性名称	数据类型	属性说明
<code>use_custom_name</code>	flag	
<code>name</code>	string	读取工作区中某个节点名称的只读属性（自动或自定义）。
<code>custom_name</code>	string	指定节点的自定义名称。
<code>tooltip</code>	string	
<code>annotation</code>	string	

属性名称	数据类型	属性说明
keywords	string	指定与对象关联的关键字列表的结构化通道（例如 ["Keyword1" "Keyword2"]）。
cache_enabled	flag	
node_type	source_supernode process_supernode terminal_supernode 为编写脚本而指定的所有节点名称	按类型引用节点的只读属性。例如，除按名称引用节点（例如 real_income）之外，还可以指定类型（例如 userinputnode 或 filternode）。

超节点属性以及所有其它节点的属性均将单独讨论。 [有关详细信息，请参阅第 281 页码第 22 章中的超节点属性。](#)

流属性

通过脚本编写可以控制多种流属性。要引用流属性，必须使用一个特定的流变量，该变量通过在流前加注一个 ^ 来标记：

```
set ^stream.execute_method = Script
```

示例

节点属性用于引用当前流中的节点。如下所示的流脚本可作为一个示例：

```
var listofnodes
var thenode
set listofnodes = ^stream.nodes

set ^stream.annotation = ^stream.annotation << "\n\nThis stream is called \" >> ^stream.name >
< \"\" and contains/ the following nodes\n"

for thenode in listofnodes
set ^stream.annotation = ^stream.annotation << "\n" << ^thenode.node_type
endfor
```

此示例使用节点属性创建了一个包含流中所有节点的列表，并将该列表写入流的注解中。生成的注解具有如下形式：

This stream is called "druglearn" and contains the following nodes

```
derivenode
neuralnetworknode
variablefilenode
typenode
c50node
filternode
```

流属性的具体说明见于下表。

属性名称	数据类型	属性说明
execute_method	Normal Script	
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY"	

属性名称	数据类型	属性说明
	"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	
date_baseline	number	
date_2digit_baseline	number	
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	flag	
import_datetime_as_string	flag	
decimal_places	number	
decimal_symbol	Default Period Comma	
angles_in_radians	flag	
use_max_set_size	flag	
max_set_size	number	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	flag	用于在流执行中自动刷新源节点。
script	string	
annotation	string	示例: set ^stream.annotation = "something interesting"

属性名称	数据类型	属性说明
name	string	示例： <code>set x = ^stream.name</code> 注意：这是一个只读属性。如果想要更改流的名称，您应该使用其它名称加以保存。
parameters		使用该属性可使您更新来自独立脚本内的流参数。 示例： <code>set ^stream.parameters.height = 23</code>
nodes		详细信息参见下方。
encoding	SystemDefault "UTF-8"	
stream_rewriting	boolean	
stream_rewriting_maximize_sql	boolean	
stream_rewriting_optimize_clem_execution	boolean	
stream_rewriting_optimize_syntax_execution	boolean	
enable_parallelism	boolean	
sql_generation	boolean	
database_caching	boolean	
sql_logging	boolean	
sql_generation_logging	boolean	
sql_log_native	boolean	
sql_log_prettyprint	boolean	
record_count_suppress_input	boolean	
record_count_feedback_interval	integer	

工程属性

许多属性都可用于工程的脚本编译。

示例

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

属性名称	数据类型	属性说明
summary	string	工程汇总—通常是注解的缩编版。
title	string	报告的标题。
author	string	报告的作者。
structure	Phase Class	通过数据挖掘阶段或对象类型（类）来确定工程的组织方式。
include_mode	IncludedItems ExcludedItems AllItems	确定工程报告中要包含项目。
select_mode	AllItems RecentItems OldItems	通过使用时间来确定要在工程报告中包含的项目。
recent_item_limit	integer	当 select_mode 为 RecentItems 时使用。
old_item_limit	integer	当 select_mode 为 OldItems 时使用。
ordering	TypeNameAdded TypeAddedName NameAddedType AddedNameType	确定项目在报告中的排列顺序。

源节点属性

源节点通用属性

所有源节点的通用属性如下所示，后面的主题是具体节点的相关信息。

示例

```
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]
```

属性名称	数据类型	属性说明
direction	Input Target Both None Partition Split Frequency RecordID	字段角色的键控属性。 用法格式： NODE.direction.FIELDNAME 注意：值 In 和 Out 现已无效。在将来的版本中可能取消对这些值的支持。
type	Range Flag Set Typeless Discrete Ordered Set Default	字段类型。如果将该属性设置为 Default，则将清除所有 values 属性设置，如果将 value_mode 设置为 Specify，则它将重新设置为 Read。如果 value_mode 已设置为 Pass 或 Read，则它将不受 type 设置的影响。 用法格式： NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	字段存储类型的只读键控属性。 用法格式： NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	字段类型和范围检查的键控属性。 用法格式： NODE.check.FIELDNAME

属性名称	数据类型	属性说明
values	[value value]	对于连续型（范围）字段而言，第一个是最小值，后一个是最大值。对于名义（集合）字段，指定所有值。对标志字段而言，第一个值代表 false，后一个值代表 true。设置该属性将自动把 value_mode 属性设置为 Specify。 用法格式： NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	确定下一次数据传递中设置某个字段值的方式。 用法格式： NODE.value_mode.FIELDNAME 注意，不能将此属性直接设置为 Specify；要使用特定值，需设置 values 属性。
default_value_mode	Read Pass	指定用默认方式设置所有字段值。 用法格式： NODE.default_value_mode 示例： set mynode.default_value_mode = Pass 该设置可以通过使用 value_mode 属性，用特定字段进行覆盖。
extend_values	flag	当 value_mode 设置为 Read 时将应用。设为 T 则将新读取的值添加到任意现有字段值。设置为 F 则丢弃现有值并添加新读取值。 用法格式： NODE.extend_values.FIELDNAME
value_labels	string	用于指定值标签。示例： set :varfilenode.value_labels.Age = [{3 three} {5 five}] 请注意，必须先指定值。
enable_missing	flag	当设置为 T 时，则激活对字段缺失值的跟踪。 用法格式： NODE.enable_missing.FIELDNAME
missing_values	[value value ...]	指定表示缺失数据的数据值。 用法格式： NODE.missing_values.FIELDNAME
range_missing	flag	当此属性设置为 T，指定是否为字段定义缺失值（空）范围。 用法格式： NODE.range_missing.FIELDNAME
missing_lower	string	当 range_missing 为真时，指定缺失值范围的下限。 用法格式： NODE.missing_lower.FIELDNAME
missing_upper	string	当 range_missing 为真时，指定缺失值范围的上限。 用法格式： NODE.missing_upper.FIELDNAME
null_missing	flag	当此属性设置为 T 时，将用 null（在本软件中显示为 \$null\$ 的未定义值）表示缺失值。 用法格式： NODE.null_missing.FIELDNAME

属性名称	数据类型	属性说明
whitespace_missing	flag	当该属性设置为 T 时，仅包含空白（空格、制表符和换行符）的值将被当成缺失值。 用法格式： NODE.whitespace_missing.FIELDNAME
description	string	用于指定字段标签或描述。
default_include	flag	用于指定默认行为是传递还是过滤字段的键控属性： NODE.default_include 示例： set mynode:filternode.default_include = false
include	flag	用于指出是包含还是过滤单个字段的键控属性： NODE.include.FIELDNAME. 示例： set mynode:filternode.include.Age = true
new_name	string	示例： set mynode:filternode.new_name.'Age' = "years"

cognosimportnode 属性



IBM Cognos BI 源节点从 Cognos BI 数据库导入数据。有关详细信息，请参阅第 2 章中的导入 Cognos 数据中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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]"}
```

cognosimportnode 属性	数据类型	属性说明
mode	Data Report	指定是导入 Cognos BI 数据（默认）还是报告。
cognos_connection	{" field", " field", ..., " field" }	包含 Cognos 服务器连接详细信息的列表属性。格式为： { " Cognos_server_URL", login_mode, " namespace", " username", " password " } 其中： Cognos_server_URL 是包含源的 Cognos 服务器的 URL login_mode 表示是否使用匿名登录，其值为 true 或 false；如果设为 true，则应将以下字段设为 "" namespace 指定用于登录服务器的安全验证提供程序

cognosimportnode 属性	数据类型	属性说明
		username 和 password 为用于登录 Cognos 服务器的用户名和密码
cognos_package_name	string	您要将数据对象导入其中的 Cognos 数据包的路径和名称, 例如: /Public Folders/GOSALES 注意: 只有正斜杠有效。
cognos_items	{ "field", "field", ..., "field" }	要导入的一个或多个数据对象的名称。字段格式为 [namespace].[query_subject].[query_item] 示例: set :cognosimport.cognos_items = {"[Inventory (query)].[Inventory].[Opening inventory]", "[Inventory (query)].[Inventory].[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	field	在导入数据之前要应用的一个或多个过滤器的名称。 示例: set :cognosimport.cognos_filters = {"[Inventory].[Filter].[MyFilter]"}
cognos_data_parameters	list	数据的提示参数值。名称和值对用大括号括起, 多个对由逗号分隔, 整个字符串由方括号括起。 格式: [{" param1", "value "}, ..., {" paramN", "value "}] 示例: set :cognosimport.cognos_data_parameters = [{"Sex-Value", "F"}, {"a", "1"}, {"b", "1"}]
cognos_report_location	field	要导入报告的文件夹或数据包的 Cognos 路径, 例如: /Public Folders/GOSALES 注意: 只有正斜杠有效。
cognos_report_name	field	要导入报告的报告位置中的路径和名称, 例如: set :cognosimport.cognos_report_name = /Jimmy/Package/Drug4nPackage/3columns

cognosimportnode 属性	数据类型	属性说明
cognos_report_parameters	list	报告参数的值。名称和值对用大括弧括起，多个对由逗号分隔，整个字符串由方括号括起。 格式： [{" param1" , "value "}, ..., {" paramN" , "value "}] 示例： set :cognosimport.cognos_re- port_parameters = [{"Sex- Value", "F"}, {"a", "1"}, {"b", "1"}]

databasenode 属性



数据库节点可用于使用 ODBC（开放数据库连接）从多种其他数据包中导入数据，这些数据包包括 Microsoft SQL Server、DB2、Oracle 等。有关详细信息，请参阅第 2 章中的数据库源节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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.eparasword
set :databasenode.tablename = ".Drug4n"
```

databasenode 属性	数据类型	属性说明
mode	Table Query	借助对话框控件，指定将 Table 连接到数据库表，或借助 SQL 来指定用 Query 查询选定数据库。
datasource	string	数据库名称（另请参阅下面的注释）。
username	string	
password	string	指定一个加密密码作为脚本中硬加密密码的备选项。 有关详细信息，请参阅第 55 页码第 5 章中的生成加密密码。此属性在执行时处于只读状态。
eparasword	string	
tablename	string	要访问的表名称。
strip_spaces	None Left Right Both	丢弃字符串中前端和尾部空格的选项。

datasourcenode 属性	数据类型	属性说明
use_quotes	AsNeeded Always Never	指定在向数据库发送查询时是否用引号括起表和列的名称（例如在包含空格或标点的情况下）。
query	string	指定要提交查询所对应的 SQL 编码。

注意：如果数据库名称（在 `datasource` 属性中）包含空格，则不会使用 `datasource`、`username` 和 `password` 的单独属性，而会使用以下格式的单个数据源属性：

datasourcenode 属性	数据类型	属性说明
datasource	string	格式： {database_name, username, password[, true false]} 最后一个参数用于加密密码。如果将其设为 <code>true</code> ，将会在使用之前对密码进行解密。

示例

```
create datasourcenode
set :datasourcenode.mode = Table
set :datasourcenode.query = "SELECT * FROM drug4n"
set :datasourcenode.datasource = {"ORA 10gR2", user1, mypsw, true}
var test_e
set test_e = :datasourcenode.eparasword
set :datasourcenode.tablename = ".Drug4n"
```

如果您要更改数据源，也可使用此格式；不过，如果您只想更改用户名或密码，则可使用 `username` 或 `password` 属性。

datacollectionimportnode 属性



IBM® SPSS® Data Collection 数据导入节点根据 IBM Corp. 市场调查产品使用的 Data Collection 数据模型导入调查数据。必须安装 Data Collection 数据库才可使用此节点。有关详细信息，请参阅第 2 章中的 Data Collection 节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

datacollectionimportnode 属性	数据类型	属性说明
metadata_name	string	MDSC 的名称。特殊值 DimensionsMDD 表示应使用标准 Data Collection 元数据文档。其他可能的值包括： mrADODsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC 特殊值 none 指示不存在 MDSC。
metadata_file	string	存储元数据的文件的名称。
casedata_name	string	CDSC 的名称。可能的值包括： mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmIDsc 特殊值 none 指示不存在 CDSC。
casedata_source_type	Unknown File Folder UDL DSN	指出 CDSC 的源类型。
casedata_file	string	当 casedata_source_type 为 File 时，则指定包含案例数据的文件。
casedata_folder	string	当 casedata_source_type 为 Folder 时，则指定包含案例数据的文件夹。
casedata_udl_string	string	当 casedata_source_type 为 UDL 时，则为包含案例数据的数据源指定 OLD-DB 连接字符串。
casedata_dsn_string	string	当 casedata_source_type 为 DSN 时，则为数据源指定 ODBC 连接字符串。
casedata_project	string	从 Data Collection 数据库中读取观测值数据时，可以输入工程的名称。对于所有其他的观测值数据类型，应将此设置留空。
version_import_mode	All Latest Specify	定义版本处理方式。
specific_version	string	当 version_import_mode 为 Specify 时，则定义要导入案例数据的版本。

datacollectionimportnode 属性	数据类型	属性说明
use_language	string	定义是否应使用指定语言的标签。
language	string	如果 use_language 的值为 True, 则定义导入时要使用的语言代码。语言代码应为案例数据中的某一可用代码。
use_context	string	定义是否应导入指定环境。环境可用于区分与响应相关的描述。
context	string	如果 use_context 的值为真, 则定义导入环境。环境应是案例数据中的某一可用环境。
use_label_type	string	定义是否应导入指定标签类型。
label_type	string	如果 use_label_type 的值为真, 则定义要导入的标签类型。标签类型应是案例数据中的某一可用标签类型。
user_id	string	对于要求显式登录的数据库, 可通过提供用户 ID 和密码来访问数据源。
password	string	
import_system_variables	Common None All	指定要导入哪些系统变量。
import_codes_variables	flag	
import_sourcefile_variables	flag	
import_multi_response	MultipleFlags Single	

excelimportnode 属性



Excel 导入节点可以从任何版本的 Microsoft Excel 中导入数据。不要求指定 ODBC 数据源。有关详细信息, 请参阅第 2 章中的 Excel 源节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
#To use a named range:
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
```

```
#To use an explicit range:
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
```

excelimportnode 属性	数据类型	属性说明
excel_file_type	Excel2003 Excel2007	
full_filename	string	完整文件名（包括路径）。
use_named_range	布尔逻辑	是否使用指定范围。如果为真，则 will use named_range 属性来指定读取范围，但忽略其它工作表和数据范围设置。
named_range	string	
worksheet_mode	Index Name	指定是否通过索引或名称来定义工作表。
worksheet_index	integer	要读取工作表的索引，开始时第一个工作表的索引为 0，第二个工作表的索引为 1，以此类推。
worksheet_name	string	要读取工作表的名称。
data_range_mode	FirstNonBlank ExplicitRange	指定确定范围的方式。
blank_rows	StopReading ReturnBlankRows	当 data_range_mode 为 FirstNonBlank 时，指定空行处理方式。
explicit_range_start	string	当 data_range_mode 为 ExplicitRange 时，指定要读取范围的起点。
explicit_range_end	string	
read_field_names	布尔逻辑	指定是否应将指定范围的第一行用作字段（列）名称。

evimportnode 属性



Enterprise View 节点用于创建指向 IBM SPSS Collaboration and Deployment Services Repository 的连接，使您可以将 Enterprise View 数据读入流中，并将模型打包装入其他用户可通过存储库访问的方案。有关详细信息，请参阅第 2 章中的 Enterprise View 节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create evimportnode
set :evimportnode.connection = ['Training data', '/Application views/Marketing', 'LATEST', 'Analytic',
    '/Data Providers/Marketing']
set :evimportnode.tablename = "cust1"
```

evimportnode 属性	数据类型	属性说明
connection	list	结构化属性 - 组成 Enterprise View 连接的参数的列表。 用法格式： evimportnode.connection = [description, app_view_path, app_view_version_label, environment, DPD_path]
tablename	string	Application View 中表格的名称。

fixedfilenode 属性



固定文件节点会从固定字段文本文件（即文件字段不定界而是从相同的位置开始且长度固定）中导入数据。机器生成的数据或遗存数据通常以固定字段格式存储。有关详细信息，请参阅第 2 章中的固定文件节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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}]
set :fixedfilenode.decimal_symbol = Period
set :fixedfilenode.lines_to_scan = 30
```

fixedfilenode 属性	数据类型	属性说明
record_len	number	指定每条记录中的字符数。
line_oriented	flag	跳过每条记录尾部的换行符。
decimal_symbol	Default Comma Period	用于数据源中的十进制分隔符的类型。示例： set :fixedfilenode.decimal_symbol = Period
skip_header	number	指定每条记录开头要忽略的行数。用于忽略列标题。
auto_recognize_datetime	flag	指定在源数据中是否自动标识日期或时间。
lines_to_scan	number	示例： set :fixedfilenode.lines_to_scan = 50.
fields	list	结构化属性。 用法格式： fixedfilenode.fields = [[field start length] {field start length}]
full_filename	string	要读取文件的全称（包括目录）。
strip_spaces	None Left Right Both	在导入时丢弃字符串中前端和尾部的空格。
invalid_char_mode	Discard Replace	从数据输入中除去无效字符（Null 值、0 或当前编码中所没有的字符），或用指定的单字符符号替换无效字符。
invalid_char_replacement	string	
use_custom_values	flag	可用通道有如下形式： set :varfilenode.use_custom_values.Age = true

fixedfilenode 属性	数据类型	属性说明
custom_storage	Unknown String Integer Real Time Date Timestamp	可用通道有如下形式： 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	该属性仅在指定自定义存储器后可用。 示例： set:varfilenode.custom 可用通道有如下形式： set :varfilenode.custom_date_format.'LaunchDate' = "DDMMYY"
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"	该属性仅在指定自定义存储器后可用。 可用通道有如下形式： set :varfilenode.custom_time_format. 'Initialize' = "HHMM"

fixedfilenode 属性	数据类型	属性说明
custom_decimal_symbol	field	仅当指定自定义存储器时可用。 可用通道有如下形式： set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	指定文本编码方法。

sasimportnode 属性



SAS 导入节点可将 SAS 数据导入到 IBM® SPSS® Modeler 中。有关详细信息，请参阅第 2 章中的 SAS 源节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

sasimportnode 属性	数据类型	属性说明
format	Windows UNIX Transport SAS7 SAS8 SAS9	要导入文件的格式。
full_filename	string	输入的完整文件名（包括路径）。
member_name	string	指定要从特定 SAS 传输文件中导入的成员。
read_formats	flag	从指定格式文件中读取数据格式（例如变量标签）。
full_format_filename	string	
import_names	NamesAndLabels LabelsasNames	指定在导入时映射变量名称和标签的方法。

statisticsimportnode 属性



IBM® SPSS® Statistics 文件节点从 SPSS Statistics 使用的 .sav 文件格式以及保存在 IBM® SPSS® Modeler 中的高速缓存文件（其也使用相同格式）读取数据。有关详细信息，请参阅第 8 章中的 Statistics 文件节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

有关此节点属性的信息，请参阅[statisticsimportnode 属性第 278 页码](#)。

user inputnode 属性



用户输入节点提供了一种用于创建综合数据的简单方式 — 可以从头开始创建也可以通过更改现有数据进行创建。此节点非常有用，例如，在希望为建模创建测试数据集时，即可使用此节点。有关详细信息，请参阅第 2 章中的用户输入节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

userinputnode 属性	数据类型	属性说明
data		可用属性有如下形式： set :userinputnode.data.Age = "1 2 3 4" 此外，字符串可以指定用逗号分隔的低值、高值和步长值。示例： set :userinputnode.data.Age = "10, 70, 5" 每个字段数据的长度可以不同，但必须与字段存储类型一致。如果为某个并不存在的字段设置值，则将创建此字段。另外，如果将某个字段的值设置为空字符串（""），则将删除指定字段。
names		设置或返回节点所生成的字段名称列表的结构化通道。 示例： ['Field1' 'Field2']
custom_storage	Unknown String Integer Real Time Date Timestamp	可用于设置或返回某个字段存储的通道。 示例： set :userinputnode.custom_storage.'Age' = "Real"
data_mode	Combined Ordered	如果指定了 Combined ，则设定值以及最小 / 最大值的每个组合都将生成一个记录。生成的记录数等于每个字段中值的数量的乘积。如果指定了 Ordered ，则从每条记录的每一列中提取一个值来生成数据行。生成的记录数等于一个与字段相关的最大数值。将为所有数据值较少的字段添加空值。
values		此属性已由 userinputnode.data 代替，今后将不再使用。

variablefilenode 属性



自由格式文件节点读取自由格式字段文本文件中的数据 — 即，其记录包含固定数量的字段，但包含不定数量字符的文件。此节点对于具有固定长度标题文本和某些特定类型注解的文件也非常有用。[有关详细信息，请参阅第 2 章中的变量文件节点中的 IBM SPSS Modeler 15 源、过程和输出节点。](#)

示例

```
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]
```

variablefilenode 属性	数据类型	属性说明
skip_header	number	指定每条记录开头要忽略的字符数。 用法格式： variablefilenode:skip_header = 3
num_fields_auto	flag	自动确定每条记录中的字段数。记录必须以换行符终止。 用法格式： variablefilenode:num_fields_auto
num_fields	number	手动指定每条记录中的字段数。
delimit_space	flag	指定文件中用于划定字段边界的字符。
delimit_tab	flag	
delimit_new_line	flag	
delimit_non_printing	flag	
delimit_comma	flag	当逗点在流中同时用作十进制分隔符和字段定界符时，将 delimit_other 设置为 true 时，然后使用 other 属性将逗号指定为定界符。
delimit_other	flag	允许使用 other 属性来指定自定义定界符。
other	string	在 delimit_other 为 true 时，指定要使用的定界符。
decimal_symbol	Default Comma Period	指定用于数据源中的十进制分隔符。
multi_blank	flag	将多个相邻空格定界符视为一个单一定界符处理。
read_field_names	flag	将数据文件的第一行作为列的标签。

variablefilenode 属性	数据类型	属性说明
strip_spaces	None Left Right Both	在导入时丢弃字符串中前端和尾部的空格。
invalid_char_mode	Discard Replace	从数据输入中除去无效字符（Null 值、0 或当前编码中所没有的字符），或用指定的单字符符号替换无效字符。
invalid_char_replacement	string	
lines_to_scan	number	指定具体数据类型的扫描行数。
auto_recognize_datetime	flag	指定在源数据中是否自动标识日期或时间。
quotes_1	Discard PairAndDiscard IncludeAsText	指定导入后单引号的处理方式。
quotes_2	Discard PairAndDiscard IncludeAsText	指定导入后双引号的处理方式。
full_filename	string	要读取的文件全称（包括目录）。
use_custom_values	flag	可用通道有如下形式： set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	可用通道有如下形式： 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	仅当指定自定义存储器时可用。 示例： set :varfilenode.custom 可用通道有如下形式： set :varfilenode.custom_date_format. 'LaunchDate' = "DDMMYY"

variablefilenode 属性	数据类型	属性说明
	q Q YYYY ww WK YYYY	
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"	仅当指定自定义存储器时可用。 可用通道有如下形式： set :varfilenode.custom_time_format. 'Initialize' = "HHMM"
custom_decimal_symbol	field	仅当指定自定义存储器时可用。 可用通道有如下形式： set :varfilenode.custom_decimal_ symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "UTF-8"	指定文本编码方法。

xmlimportnode 属性



XML 源节点将 XML 格式的数据导入到流中。可以导入某个目录中的单个文件或所有文件。还可选择指定架构文件，以从中读取 XML 结构。有关详细信息，请参阅第 2 章中的 XML 源节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create xmlimportnode
set :xmlimportnode.full_filename = "c:\import\ebooks.xml"
set :xmlimportnode.records = "/author/name"
```

xmlimportnode 属性	数据类型	属性说明
read	single directory	读取单个数据文件（默认），或目录中的所有 XML 文件。
recurse	flag	指定是否另外读取指定目录的所有子目录中的 XML 文件。
full_filename	string	（必需）要导入的 XML 文件的完整路径和文件名（如果 read = single）。
directory_name	string	（必需）要从中导入 XML 文件的目录的完整路径和名称（如果 read = directory）。

xmlimportnode 属性	数据类型	属性说明
full_schema_filename	string	要从中读取 XML 结构的 XSD 或 DTD 文件的完整路径和文件名。如果您省略了此参数，将从 XML 源文件中读取结构。
records	string	XPath 表达式（例如，/author/name），用以定义记录边界。每次在源文件中遇到此元素时，都将创建新的记录。
mode	read specify	读取所有数据（默认），或指定要读取的项目。
fields		要导入的项目（元素和属性）列表。列表中的每项为 XPath 表达式。

记录操作节点属性

appendnode 属性



“附加”节点连接各组记录。也可以用于将数据集与结构类似但内容不同的数据合并起来。有关详细信息，请参阅第 3 章中的追加节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

appendnode 属性	数据类型	属性说明
match_by	Position Name	可以根据字段在主数据源中的位置或输入数据集中字段的名称来附加数据集。
match_case	flag	匹配字段名称时启用区分大小写。
include_fields_from	Main All	
create_tag_field	flag	
tag_field_name	string	

aggregatenode 属性



“合计”节点用汇总和合计的输出记录替代一系列输入记录。有关详细信息，请参阅第 3 章中的汇总节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create aggregatenode
connect :databasenode to :aggregatenode
set :aggregatenode.contiguous = True
set :aggregatenode.keys = ['Drug']
set :aggregatenode.aggregates.Age = [Sum Mean]
set :aggregatenode.inc_record_count = True
set :aggregatenode.count_field = "index"
set :aggregatenode.extension = "Aggregated_"
```

```
set :aggregatenode.add_as = Prefix
```

aggregatenode 属性	数据类型	属性说明
keys	[field field ... field]	列出可用作汇总的关键字段。例如，如果 Sex 和 Region 是关键字段，M 和 F 与区域 N 和 S 的每个唯一性组合（四个唯一性组合）都将具有一个经过汇总的记录。
contiguous	flag	如果您知道在输入中具有相同关键值的所有记录被分成了一组，则可以选择此选项（例如，如果对关键字段上的输入进行了排序）。这样做有助于提高性能。
aggregates		一种结构化属性，它列出其值将被汇总的数字字段以及选定的汇总模式。示例： set :aggregatenode. aggregates.Age = [Sum Mean Min Max SDev Median Count Variance Firstquartile Thirdquartile]，其中所需汇总方法包括在列表中。
extension	string	为重复汇总字段指定前缀或后缀（样本如下）。
add_as	Suffix Prefix	
inc_record_count	flag	创建一个额外字段，该字段指定为形成每条汇总记录汇总了多少条输入记录。
count_field	string	指定记录计数字段的名称。

balancenode 属性



“平衡”节点纠正数据集中的不平衡，因而它遵循指定的条件。“平衡”指定调整根据指定系数条件为真的记录的比例。[有关详细信息，请参阅第 3 章中的平衡节点中的 IBM SPSS Modeler 15 源、过程和输出节点。](#)

示例

```
create balancenode
set :balancenode.training_data_only = true
set :balancenode.directives = \
  [{1.3 "Age > 60"} {1.5 "Na > 0.5"}]
```

balancenode 属性	数据类型	属性说明
directives		根据指定数字平衡字段值比例的结构化属性（参阅下面的示例）。
training_data_only	flag	指定仅平衡训练数据。如果流中不存在分区字段，则忽略该选项。

示例

```
create balancenode
set :balancenode.directives = \
  [{1.3 "Age > 60"} {1.5 "Na > 0.5"}]
```

此节点属性使用以下格式：

[{ 数字 字符串 } \ { 数字 字符串 } \ ... { 数字 字符串 }]。

注意：如果在表达式中嵌入字符串（用半角双引号括起来），则需要在字符串前添加转义字符“\”。“\”字符同时也是行继续符，通过它可将参数对齐以使参数清晰明了。

distinctnode 属性



区分节点会除去重复的记录，方法是：将第一个可区分记录传递到数据流，或丢弃第一个记录而将任何重复记录传递到数据流。[有关详细信息，请参阅第 3 章中的区分节点中的 IBM SPSS Modeler 15 源、过程和输出节点。](#)

示例

```
create distinctnode
set :distinctnode.mode = Include
set :distinctnode.fields = ['Age' 'Sex']
set :distinctnode.keys_pre_sorted = True
```

distinctnode 属性	数据类型	属性说明
mode	Include Discard	既可以将第一条区分记录包括在数据流中，也可以丢弃第一条区分记录并将任何重复记录传送到数据流。
fields	[field field field]	列出用于确定记录是否相同的字段。
low_distinct_key_count	flag	指定您只具有少量记录和/或少量关键字字段唯一值。
keys_pre_sorted	flag	指定具有相同键值的所有记录在输入中分组在一起。

mergenode 属性



合并节点获取多个输入记录并创建包含某些或全部输入字段的单个输出记录。这对于合并来源不同的数据非常有用，例如内部客户数据和已购买人群统计数据。[有关详细信息，请参阅第 3 章中的合并节点中的 IBM SPSS Modeler 15 源、过程和输出节点。](#)

示例

```

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}]

```

mergenode 属性	数据类型	属性说明
method	Order Keys Condition	指定记录是否按它们在数据文件中的罗列顺序进行合并，是否使用一个或多个关键字段来合并并在关键字段中具有相同值的记录，或者是否在满足指定条件时合并记录。
condition	string	如果 method 设置为 Condition，指定包括或丢弃记录的条件。
key_fields	[field field field]	
common_keys	flag	
join	Inner FullOuter PartialOuter Anti	示例如下： set :merge.join = FullOuter
outer_join_tag.n	flag	在此属性中，n 是“选择数据集”对话框中显示的标签名。注意，可以指定多个标签名，因为任何数量的数据集都无法提供完整记录。
single_large_input	flag	指定是否进行优化，以使一个输入与其他输入相比具有一个相对较大的输入值。
single_large_input_tag	string	按“选择较大数据集”对话框中的显示指定标签名。请注意，该属性的用法与 outer_join_tag 属性的用法略有不同（标记与字符串），因为前者只能指定一个输入数据集。
use_existing_sort_keys	flag	指定输入值是否已根据一个或多个关键字段进行排序。
existing_sort_keys	[[{string Ascending} \ {string Descending}]	指定已排序的字段及其排序方向。

rfmaggregatenode 属性



使用“近因、频数和货币 (RFM) 汇总”节点，您可以采用客户的历史交易数据，删除所有无用数据以及将所有他们保留的交易数据组合成一行，且该行中列出了他们与您上次谈业务的时间、所完成的交易量以及这些交易的总货币价值。有关详细信息，请参阅第 3 章中的 RFM 汇总节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

rfmaggregatenode 属性	数据类型	属性说明
relative_to	Fixed Today	指定计算交易近因的日期。
reference_date	日期	仅在 relative_to 中选择 Fixed 时才可用。
contiguous	flag	如果您的数据进行了预先排序，以便所有 ID 相同的记录一起出现在数据流中，那么选择此选项可以加快处理速度。
id_field	field	指定该字段以用来识别客户及其交易。
date_field	field	指定将要用来计算近因的日期字段。
value_field	field	指定该字段以用来计算货币值。
extension	string	为重复汇总字段指定前缀或后缀。
add_as	Suffix Prefix	指定是否应作为前缀或后缀来添加 extension。
discard_low_value_records	flag	启用使用 discard_records_below 设置。
discard_records_below	number	可在计算 RFM 总计时，指定一个最小值，凡低于该值的交易详细信息都不再被使用。值单位与所选 value 字段相关。
only_recent_transactions	flag	启用使用 specify_transaction_date 或 transaction_within_last 设置。
specify_transaction_date	flag	
transaction_date_after	日期	只有选中 specify_transaction_date 时才可用。指定交易日期以在分析时包含其之后的记录。

rfmaggregatenode 属性	数据类型	属性说明
transaction_within_last	number	只有选中 <code>transaction_within_last</code> 时才可用。指定从计算相对于以下内容的近因日期字段所返回的周期数和周期类型（天、周、月或年），在此日期之后的记录将被包含在您的分析中。
transaction_scale	Days Weeks Months Years	只有选中 <code>transaction_within_last</code> 时才可用。指定从计算相对于以下内容的近因日期字段所返回的周期数和周期类型（天、周、月或年），在此日期之后的记录将被包含在您的分析中。
save_r2	flag	显示每个客户第二个最近交易的日期。
save_r3	flag	只有选中 <code>save_r2</code> 时才可用。显示每个客户第三个最近交易的日期。

samplenode 属性



样本节点选择记录的子集。受支持的样本类型有许多，其中包括分层、聚类和随机（结构化）样本。取样对于提高性能和选择相关记录组或交易组用于分析会很有用。有关详细信息，请参阅第 3 章中的样本节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
/* Create two Sample nodes to extract
   different samples from the same data */

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

set node = create samplenode at 300 100
rename ^node as 'First 500'
connect :variablefilenode to 'First 500'
set 'First 500':samplenode.method = Simple
set 'First 500':samplenode.mode = Include
set 'First 500':samplenode.sample_type = First
set 'First 500':samplenode.first_n = 500

set node = create samplenode at 300 200
rename ^node as 'Custom Strata'
connect :variablefilenode to 'Custom Strata'
set 'Custom Strata':samplenode.method = Complex
set 'Custom Strata':samplenode.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"}]
```

samplenode 属性	数据类型	属性说明
method	Simple Complex	
mode	Include Discard	包括或丢弃满足指定条件的记录。
sample_type	First OneInN RandomPct	指定抽样方法。示例如下： set :samplenode.sample_type = First set :samplenode.first_n = 100
first_n	integer	将包括或丢弃直到指定截止点的记录。
one_in_n	number	每隔 n-1 条记录包括或丢弃一条记录。
rand_pct	number	指定要包括或丢弃记录的百分比。
use_max_size	flag	启用使用 maximum_size 设置。
maximum_size	integer	指定要包括在数据流中或丢弃的最大样本量。此选项是冗余选项，因此指定 First 和 Include 时会被禁用。
set_random_seed	flag	启用随机种子设置。
random_seed	integer	指定用作随机种子的值。
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	number	
fixed_counts	integer	
variable_proportions	field	
variable_counts	field	
use_min_stratum_size	flag	
minimum_stratum_size	integer	仅当抽取复杂样本 Sample units=Proportions 时才应用此选项。
use_max_stratum_size	flag	
maximum_stratum_size	integer	仅当抽取复杂样本 Sample units=Proportions 时才应用此选项。
clusters	field	
stratify_by	[field1 ... fieldN]	
specify_input_weight	flag	
input_weight	field	
new_output_weight	string	
sizes_proportions	[[{stringstring value}{stringstring value}...]]	如果 sample_units=proportions 且 sample_size_proportions=Custom, 指定层字段值每个可能组合的值。

samplenode 属性	数据类型	属性说明
default_proportion	number	
sizes_counts	[[{stringstring value}{stringstring value}...]]	指定层字段值每个可能的组合值。用法与 sizes_proportions 的用法相似，但指定的是整数，而非比例。
default_count	number	

selectnode 属性



选择节点可基于特定条件从数据流中选择或丢弃记录子集。例如，可以选择有关特定销售区域的记录。有关详细信息，请参阅第 3 章中的选择节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create selectnode
set :selectnode.mode = Include
set :selectnode.condition = "Age < 18"
```

selectnode 属性	数据类型	属性说明
mode	Include Discard	指定是包括还是丢弃选定记录。
condition	string	包括或丢弃记录的条件。

sortnode 属性



排序节点可根据一个或多个字段的值将记录按升序或降序排序。有关详细信息，请参阅第 3 章中的排序节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create sortnode
set :sortnode.keys = [{"Age" Ascending} {"Sex" Descending}]
set :sortnode.default_ascending = False
set :sortnode.use_existing_keys = True
set :sortnode.existing_keys = [{"Age" Ascending}]
```

sortnode 属性	数据类型	属性说明
keys	[[{string Ascending} \ {string Descending}]]	指定要依据其进行排序的字段（参阅下面的示例）。如果未指定方向，则会使用默认值。
default_ascending	flag	指定默认排序顺序。
use_existing_keys	flag	指定是否使用以前已排序字段的排序顺序来优化现在的排序。
existing_keys		指定已排序的字段及其排序方向。使用的格式与 keys 属性相同。

字段操作节点属性

anonymizenode 属性



匿名化节点用于转换字段名和字段值的下游表示方式，从而掩饰了原始数据。如果要允许其他用户使用敏感数据（如客户名称或其他详细信息）构建模型，这种节点将十分有用。有关详细信息，请参阅第 4 章中的匿名化节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create anonymizenode
set:anonymizenode.enable_anonymize = age
set:anonymizenode.use_prefix = true
set:anonymizenode.prefix = "myprefix"
set:anonymizenode.transformation = Random
set:anonymizenode.set_random_seed = true
set:anonymizenode.random_seed = "123"
```

anonymizenode 属性	数据类型	属性说明
enable_anonymize	flag	设置为 T 时，可激活匿名化字段值（相当于在“匿名值”列中为该字段选择是）。
use_prefix	flag	设置为 T 时，如果已指定自定义前缀，则会使用该前缀。适用于将通过杂凑法被匿名化的字段，而且相当于在“替换值”对话框中为该字段选择自定义单选按钮。
prefix	string	相当于在“替换值”对话框中将前缀键入文本框。如果未指定其他任何值，则默认前缀即是该默认值。
transformation	Random Fixed	确定通过变换法匿名化的字段的变换参数是随机的还是固定的。
set_random_seed	flag	设置为 T 时，将使用指定的种子值（如果变换也设置为 Random）。
random_seed	integer	当 set_random_seed 设置为 T 时，该值是随机数的种子。
scale	number	当变换设置为“固定”时，该值用于“变换尺度”。通常，最大尺度值为 10，但可能会被减小以避免溢出。
translate	number	当变换设置为“固定”时，该值用于“变换”。通常，最大变换值为 1000，但可能会被减小以避免溢出。

autodataprenode 属性



自动数据准备 (ADP) 节点可分析您的数据并标识修正, 筛选出存在问题或可能无用的字段, 并在适当的情况下派生新的属性, 通过智能筛选和抽样技术改进性能。您可以完全自动化地使用节点, 允许节点选择并应用修正, 或者也可在修正前预览更改, 按照需要接受、拒绝或修改。有关详细信息, 请参阅第 4 章中的自动数据准备中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create autodataprenode
set:autodataprenode.objective = Balanced
set:autodataprenode.excluded_fields = Filter
set:autodataprenode.prepare_dates_and_times = true
set:autodataprenode.compute_time_until_date = true
set:autodataprenode.reference_date = Today
set:autodataprenode.units_for_date_durations = Automatic
```

autodataprenode 属性	数据类型	属性说明
objective	Balanced Speed Accuracy Custom	
custom_fields	flag	如果为 true, 则允许您指定当前节点的目标、输入和其他字段。如果为 false, 则使用来自上游类型节点的当前设置。
target	field	指定一个目标字段。
inputs	[field1 ... fieldN]	模型使用的输入或预测变量字段。
use_frequency	flag	
frequency_field	field	
use_weight	flag	
weight_field	field	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	flag	控制对所有日期与时间字段的访问
compute_time_until_date	flag	
reference_date	Today Fixed	
fixed_date	日期	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	flag	

autodatapreinode 属性	数据类型	属性说明
reference_time	CurrentTime Fixed	
fixed_time	时间	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	flag	
extract_month_from_date	flag	
extract_day_from_date	flag	
extract_hour_from_time	flag	
extract_minute_from_time	flag	
extract_second_from_time	flag	
exclude_low_quality_inputs	flag	
exclude_too_many_missing	flag	
maximum_percentage_missing	number	
exclude_too_many_categories	flag	
maximum_number_categories	number	
exclude_if_large_category	flag	
maximum_percentage_category	number	
prepare_inputs_and_target	flag	
adjust_type_inputs	flag	
adjust_type_target	flag	
reorder_nominal_inputs	flag	
reorder_nominal_target	flag	
replace_outliers_inputs	flag	
replace_outliers_target	flag	
replace_missing_continuous_inputs	flag	
replace_missing_continuous_target	flag	
replace_missing_nominal_inputs	flag	
replace_missing_nominal_target	flag	
replace_missing_ordinal_inputs	flag	
replace_missing_ordinal_target	flag	
maximum_values_for_ordinal	number	
minimum_values_for_continuous	number	
outlier_cutoff_value	number	
outlier_method	Replace Delete	
rescale_continuous_inputs	flag	

autodatapreinode 属性	数据类型	属性说明
rescaling_method	MinMax ZScore	
min_max_minimum	number	
min_max_maximum	number	
z_score_final_mean	number	
z_score_final_sd	number	
rescale_continuous_target	flag	
target_final_mean	number	
target_final_sd	number	
transform_select_input_fields	flag	
maximize_association_with_target	flag	
p_value_for_merging	number	
merge_ordinal_features	flag	
merge_nominal_features	flag	
minimum_cases_in_category	number	
bin_continuous_fields	flag	
p_value_for_binning	number	
perform_feature_selection	flag	
p_value_for_selection	number	
perform_feature_construction	flag	
transformed_target_name_extension	string	
transformed_inputs_name_extension	string	
constructed_features_root_name	string	
years_duration_name_extension	string	
months_duration_name_extension	string	
days_duration_name_extension	string	
hours_duration_name_extension	string	
minutes_duration_name_extension	string	
seconds_duration_name_extension	string	
year_cyclical_name_extension	string	
month_cyclical_name_extension	string	
day_cyclical_name_extension	string	
hour_cyclical_name_extension	string	

autodatapreprenode 属性	数据类型	属性说明
minute_cyclical_name_extension	string	
second_cyclical_name_extension	string	

binningnode 属性



分箱节点根据一个或多个现有连续（数值范围）字段的值自动创建新的名义（集合）字段。例如，用户可将连续收入字段转换为一个包含各组收入的新的分类字段，作为其与平均值之间的偏差。一旦创建新字段分级后，即可根据割点创建“衍生”节点。有关详细信息，请参阅第 4 章中的分级节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

binningnode 属性	数据类型	属性说明
fields	[field1 field2 ... fieldn]	待转换的连续（数值范围）字段。可以同时为多个字段进行分级。
method	FixedWidth EqualCount Rank SDev Optimal	用于为新字段分级（类别）确定分割点的方法。
recalculate_bins	Always IfNecessary	指定是重新计算分级且每次执行节点时将数据放置在相关分级中，还是仅将数据添加到现有分级和已添加的新分级中。
fixed_width_name_extension	string	默认扩展名为 _BIN。
fixed_width_add_as	Suffix Prefix	指定是将扩展名添加到字段名的末尾（后缀）还是开头（前缀）。默认扩展名为 income_BIN。
fixed_bin_method	Width Count	
fixed_bin_count	integer	指定用于确定新字段的固定宽度分级（类别）数的整数。
fixed_bin_width	real	用于计算分级宽度的值（整数或实数）。
equal_count_name_extension	string	默认扩展名为 _TILE。

binningnode 属性	数据类型	属性说明
equal_count_add_as	Suffix Prefix	指定用于使用标准 p-tile 法生成的字段名的扩展名（后缀或前缀）。默认扩展名为 _TILE 加上 N，其中 N 是分位数。
tile4	flag	生成四分位数分级，每个分级中包含 25% 的观测值。
tile5	flag	生成五分位数分级。
tile10	flag	生成十分位数分级。
tile20	flag	生成二十分位数分级。
tile100	flag	生成百分位数分级。
use_custom_tile	flag	
custom_tile_name_extension	string	默认扩展名为 _TILEN。
custom_tile_add_as	Suffix Prefix	
custom_tile	integer	
equal_count_method	RecordCount ValueSum	RecordCount 方法是每个分级分配相同数目的记录，而 ValueSum 方法是使分配记录后每个分级中值的总和相等。
tied_values_method	Next Current Random	指定要输入的分级结值数据。
rank_order	Ascending Descending	此属性包括 Ascending（最低值标记为 1）或 Descending（最高值标记为 1）。
rank_add_as	Suffix Prefix	此选项适用于排序、分数排序和百分比排序。
rank	flag	
rank_name_extension	string	默认扩展名为 _RANK。
rank_fractional	flag	排序观测值，其中新字段的值是排序值除以非缺失观测值的权重和。分数排序值介于 0 - 1 之间。
rank_fractional_name_extension	string	默认扩展名为 _F_RANK。
rank_pct	flag	每个排序值除以具有有效值的记录数然后乘以 100。百分比分数排序值介于 1 - 100 之间。
rank_pct_name_extension	string	默认扩展名为 _P_RANK。
sdev_name_extension	string	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	string	默认扩展名为 _OPTIMAL。
optimal_add_as	Suffix Prefix	

binningnode 属性	数据类型	属性说明
optimal_supervisor_field	field	作为监督字段选择的字段，为分级选择的字段与之相关。
optimal_merge_bins	flag	指定将所有具有较小观测值计数的分级添加到更大的相邻分级。
optimal_small_bin_threshold	integer	
optimal_pre_bin	flag	表示要进行数据集的预分级。
optimal_max_bins	integer	指定上限以避免创建过大分级数。
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded Bounded	

derivenode 属性



导出节点将修改数据值或根据一个或多个现有字段创建新字段。它可创建的字段类型包括公式、标志、名义、状态、计数和条件。有关详细信息，请参阅第 4 章中的派生节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
# Create and configure a Flag Derive field node
create divenode
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"
# Create and configure a Conditional Derive field node
create divenode
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\`"
```

derivenode 属性	数据类型	属性说明
new_name	string	新字段的名称。
mode	Single Multiple	指定单个或多个字段。
fields	[field field field]	仅用于在 Multiple 模式下选择多个字段。
name_extension	string	为新字段名指定扩展名。
add_as	Suffix Prefix	将扩展名添加为字段名的前缀（开头）或后缀（末尾）。

derivnode 属性	数据类型	属性说明
result_type	Formula Flag Set State Count Conditional	可创建的六种类型的新字段。
formula_expr	string	用于在导出节点计算新字段值的表达式。
flag_expr	string	
flag_true	string	
flag_false	string	
set_default	string	
set_value_cond	string	提供与给定值相关联的条件的结构化属性。 用法格式： set :derivnode. set_value_cond. Retired = 'age > 65'
state_on_val	string	指定满足 On 条件时新字段的值。
state_off_val	string	指定满足 Off 条件时新字段的值。
state_on_expression	string	
state_off_expression	string	
state_initial	On Off	为新字段的每个记录分配初始值 On 或 Off。可在满足每个条件时更改此值。
count_initial_val	string	
count_inc_condition	string	
count_inc_expression	string	
count_reset_condition	string	
cond_if_cond	string	
cond_then_expr	string	
cond_else_expr	string	

ensemblenode 属性



整体节点可结合使用两个或多个模型块，这样所获得的预测会比通过任意一个模型获得的预测更为准确。有关详细信息，请参阅第 4 章中的整体节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
# Create and configure an Ensemble node
# Use this node with the models in 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 :ensembledenode.flag_voting_tie_selection = HighestConfidence
```

ensembledenode 属性	数据类型	属性说明
ensemble_target_field	field	为在整体中使用的所有模型指定目标字段。
filter_individual_model_output	flag	指定是否应抑制各个模型的评分结果。
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	指定用于确定整体得分的方法。仅当选定的目标为标志字段时，才会应用该设置。
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	指定用于确定整体得分的方法。仅当选定的目标为名义字段时，才会应用该设置。
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	如果已选定投票方法，则指定解决结的方法。仅当选定的目标为标志字段时，才会应用该设置。
set_voting_tie_selection	Random HighestConfidence	如果已选定投票方法，则指定解决结的方法。仅当选定的目标为名义字段时，才会应用该设置。
calculate_standard_error	flag	如果目标字段是连续的，则默认情况下会运行标准误差计算以计算测量或估算值与真值之间的差值；并显示这些估算值的相近匹配程度。

fillernode 属性



“填充”节点会替换字段值并更改存储。您可以选择基于 CLEM 条件（例如 @BLANK(@FIELD)）的替换值。或者，也可以选择将所有空值或 Null 值替换为特定值。“填充”节点经常结合“类型”节点使用以替换缺失值。有关详细信息，请参阅第 4 章中的填充节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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\'"
```

fillernode 属性	数据类型	属性说明
fields	[field field field]	数据集中其值将被检查并替换的字段。
replace_mode	Always Conditional Blank Null BlankAndNull	可以替换所有值、空白值或空值，也可以根据指定条件进行替换。
condition	string	
replace_with	string	

filternode 属性



过滤节点用于在源节点之间过滤（丢弃）字段，对字段进行重命名和映射。有关详细信息，请参阅第 4 章中的过滤或重命名字段中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create filternode
set :filternode.default_include = True
set :filternode.new_name.'Drug' = 'Chemical'
set :filternode.include.'Drug' = off
```

使用 default_include 属性。注意，设置 default_include 属性的值不会自动包括或排除所有字段；它只确定针对当前选定字段的默认行为。这在功能上等效于单击“过滤节点”对话框中的默认情况下包括字段按钮。例如，假设运行以下脚本：

```
set Filter.default_include=False
# Include only fields in the list
for f in Age Sex
  set Filter.include.^f=True
endfor
```

这会使节点传递字段年龄和性别，而丢弃其他所有字段。现在，假设再次运行相同脚本但指定两个不同字段：

```
set Filter.default_include=False
# Include only fields in the list
for f in BP Na
  set Filter.include.^f=True
endfor
```

此时会在过滤器中再添加两个字段，因此总共传递四个字段（年龄、性别、BP、Na）。换句话说，将 default_include 的值重新设置为 False 不会自动重新设置所有字段。

此外，如果现在通过使用脚本或在“过滤节点”对话框中将 `default_include` 的值更改为 `True`，则会使此行为发生颠倒，即，将丢弃而非包括上面列出的四个字段。如果有疑问，可使用“过滤节点”对话框中的控件进行实验，这将有助于理解此交互效应。

filternode 属性	数据类型	属性说明
<code>default_include</code>	flag	用于指定默认行为是传递还是过滤字段的键控属性： <code>NODE.include.FIELDNAME</code> 示例如下： <code>set mynode:filternode.default_include = false</code> 注意，设置此属性不会自动包括或排除所有字段；它只确定默认情况下是包括还是排除选定字段。有关其他注释请参阅下面的示例。
<code>include</code>	flag	用于包括和删除字段的键控属性。 用法格式： <code>NODE.include.FIELDNAME</code> 示例如下： <code>set mynode:filternode.include.Age = false</code>
<code>new_name</code>	string	示例如下： <code>set mynode:filternode.new_name.Age = "age"</code>

historynode 属性



历史节点将创建新字段，其中包含之前记录中的字段数据。历史节点最常用于顺序数据，如时间序列数据。使用历史节点前，您可能想用排序节点对此数据进行排序。有关详细信息，请参阅第 4 章中的历史节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create historynode
set :historynode.fields = ['Drug']
set :historynode.offset = 1
set :historynode.span = 3
set :historynode.unavailable = Discard
set :historynode.fill_with = "undef"
```

historynode 属性	数据类型	属性说明
<code>fields</code>	[field field field]	需要其历史的字段。
<code>offset</code>	number	指定要从中提取历史字段值的最新记录（当前记录之前的记录）。
<code>span</code>	number	指定要从中提取值的以前记录的数目。

historynode 属性	数据类型	属性说明
unavailable	Discard Leave Fill	处理不含历史值的记录时，通常参考没有以前的记录作为历史的前几个记录（位于数据集顶部）。
fill_with	String Number	指定要用于无历史值可用的记录的值或字符串。

partitionnode 属性



分区节点可生成分区字段，该字段可将数据分割为单独的子集以便在模型构建的训练、测试和验证阶段使用。有关详细信息，请参阅第 4 章中的分区节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

partitionnode 属性	数据类型	属性说明
new_name	string	由节点生成的分区字段的名称。
create_validation	flag	指定是否应创建验证分区。
training_size	integer	要分配给训练分区的记录的百分比 (0 - 100)。
testing_size	integer	要分配给测试分区的记录的百分比 (0 - 100)。
validation_size	integer	要分配给验证分区的记录的百分比 (0 - 100)。如果未创建验证分区，则忽略此属性。
training_label	string	训练分区的标签。
testing_label	string	测试分区的标签。
validation_label	string	验证分区的标签。如果未创建验证分区，则忽略此属性。
value_mode	System SystemAndLabel Label	指定用于表示数据中每个分区的值。例如，训练样本可以表示为系统整数 1、标签 Training 或二者的组合 1_Training。
set_random_seed	布尔逻辑	指定是否应使用用户指定的随机种子。
random_seed	integer	用户指定的随机种子值。如果要使用此值，set_random_seed 必须设置为 True。

partitionnode 属性	数据类型	属性说明
enable_sql_generation	布尔逻辑	指定是否使用 SQL 回送以分配记录到分区。
unique_field		指定输入字段，用以确保以随机但可重复的方式将记录分配到分区。如果要使用此值，enable_sql_generation 必须设置为 True。

reclassifynode 属性



重新分类节点可将一组分类值转换为另一组值。对于压缩类别或为分析而进行的数据重新分组，重新分类非常有用。有关详细信息，请参阅第 4 章中的重新对节点分类中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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]
```

reclassifynode 属性	数据类型	属性说明
mode	Single Multiple	Single 可为一个字段进行重新分类。Multiple 可激活选项，启用对一个以上字段的一次性类别转换。
replace_field	flag	
field	string	仅在 Single 模式下使用。
new_name	string	仅在 Single 模式下使用。
fields	[field1 field2 ... fieldn]	仅在 Multiple 模式下使用。
name_extension	string	仅在 Multiple 模式下使用。
add_as	Suffix Prefix	仅在 Multiple 模式下使用。
reclassify	string	字段值的结构化属性。 用法格式： NODE.reclassify. OLD_VALUE 示例如下： set :reclassifynode.reclassifify.'drugB' = 'Yes'
use_default	flag	使用默认值。

reclassifynode 属性	数据类型	属性说明
default	string	指定默认值。
pick_list	[string string ... string]	允许用户导入已知新值的列表以填充表中的下拉列表。 示例如下： set :reclassify.pick_list = [fruit dairy cereals]

reordernode 属性



字段重排节点定义了用于显示下游字段的自然顺序。此顺序将影响字段在多个位置的显示方式，如表格、列表和字段选择器。处理大型数据集时，此操作有助于使所需字段更为直观。有关详细信息，请参阅第 4 章中的字段重排节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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]
```

reordernode 属性	数据类型	属性说明
mode	Custom Auto	可以自动对值进行排序，也可以指定自定义顺序。
sort_by	Name Type Storage	
ascending	flag	
start_fields	[field1 field2 ... fieldn]	新字段插入到这些字段之后。
end_fields	[field1 field2 ... fieldn]	新字段插入到这些字段之前。

restructurenode 属性



重新构建节点可将一个名义字段或标志字段转换为一组字段（该字段组由已成为另一字段的值填充）。例如，给定一个名为支付类型的字段，其值为贷方、现金和借方，则将创建三个新字段（贷方、现金、借方），每个字段可能包含实际支付的值。有关详细信息，请参阅第 4 章中的重新结构化节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"]
```

restructurenode 属性	数据类型	属性说明
fields_from	[category category category] all	例如， set :restructurenode.fields_from.Drug = [drugA drugB] 可创建名为 Drug_drugA 和 Drug_drugB 的字段。 使用指定字段的所有类别： set :restructurenode.fields_from.Drug = all
include_field_name	flag	指示是否在重新结构化的字段名中使用字段名。
value_mode	OtherFields Flags	表示用于为重新结构化字段指定值的模式。如果选择 OtherFields，必须指定要使用哪些字段（参阅下文）。如果选择 Flags，则值为数值标志。
value_fields	[field field field]	如果 value_mode 为 OtherFields，则为必需。指定使用哪些字段作为值字段。

rfmanalysisnode 属性



通过近因、频数和货币（RFM）分析节点，您可以检查客户最近一次购买您产品或服务的时间（近因）、客户购买的频率（频数）以及客户支付的所有交易金额（货币），确定可能成为最佳客户的数量。有关详细信息，请参阅第 4 章中的 RFM 分析节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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]
```

rfmanalysisnode 属性	数据类型	属性说明
recency	field	指定近因字段。它有可能是日期、时间戳或简单的数值。
frequency	field	指定频数字段。
monetary	field	指定货币字段。
recency_bins	integer	指定要生成的近因分级数量。
recency_weight	number	指定应用于近因数据的权重。默认值为 100。
frequency_bins	integer	指定要生成的频数分级数量。

rfanalysisnode 属性	数据类型	属性说明
frequency_weight	number	指定应用于频数数据的权重。默认值为 10。
monetary_bins	integer	指定要生成的货币分级数量。
monetary_weight	number	指定应用于货币数据的权重。默认值为 1。
tied_values_method	Next Current	指定要输入的分级结值数据。
recalculate_bins	Always IfNecessary	
add_outliers	flag	仅当 recalculate_bins 设置为 IfNecessary 时可用。如果已设置，则将位于下限分级以下的记录添加到下限分级中，并且将最高分级以上的记录添加到最高分级中。
binned_field	Recency Frequency Monetary	
recency_thresholds	value value	仅当 recalculate_bins 设置为 Always 时可用。指定近因分级的下限阈值和上限阈值。一个分级的上限阈值用作下一个分级的下限阈值—例如，[10 30 60] 可定义两个分级，第一个分级的上限阈值和下限阈值分别为 10 和 30，第二个分级的两个阈值分别为 30 和 60。
frequency_thresholds	value value	仅当 recalculate_bins 设置为 Always 时可用。
monetary_thresholds	value value	仅当 recalculate_bins 设置为 Always 时可用。

settoflagnode 属性



“设为标志”节点根据为一个或多个名义字段定义的分类值获取多个标志字段。有关详细信息，请参阅第 4 章中的设为标志节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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']
```

settoflagnode 属性	数据类型	属性说明
fields_from	[category category category] all	例如， set :settoflagnode.fields_from.Drug = [drugA drugB] 创建名为 Drug_drugA 和 Drug_drugB 的标志字段。 使用指定字段的所有类别： set :settoflagnode.fields_from.Drug = all
true_value	string	指定设置标志时节点所使用的真值。默认值为 T。
false_value	string	指定设置标志时节点所使用的假值。默认值为 F。
use_extension	flag	使用扩展名作为新标志字段的后缀或前缀。
extension	string	
add_as	Suffix Prefix	指定所添加的扩展名是后缀还是前缀。
aggregate	flag	根据关键字段将记录分组。如果有任何记录被设置为真，则会启用组中的所有标志字段。
keys	[field field field]	关键字段。

statisticstransformnode 属性



Statistics 转换节点针对 IBM® SPSS® Modeler 中的数据源运行所选的 IBM® SPSS® Statistics 语法命令。此节点需要 SPSS Statistics 的许可副本。
有关详细信息，请参阅第 8 章中的 Statistics 转换节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

有关此节点属性的信息，请参阅 [statisticstransformnode 属性第 278 页码](#)。

timeintervalnode 属性



时间区间节点指定区间，并创建用于对时间序列数据进行建模的标签（如果需要）。如果值的间隔不是均匀的，则此节点会根据需要填充值或将值集中起来以生成均匀的记录区间。有关详细信息，请参阅第 4 章中的时间区间节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create timeintervalnode
set :timeintervalnode.interval_type=SecondsPerDay
set :timeintervalnode.days_per_week=4
set :timeintervalnode.week_begins_on=Tuesday
set :timeintervalnode.hours_per_day=10
set :timeintervalnode.day_begins_hour=7
```

```

set :timeintervalnode. day_begins_minute=5
set :timeintervalnode. day_begins_second=17
set :timeintervalnode. mode=Label
set :timeintervalnode. year_start=2005
set :timeintervalnode. month_start=January
set :timeintervalnode. day_start=4
set :timeintervalnode. pad. AGE=MeanOfRecentPoints
set :timeintervalnode. agg_mode=Specify
set :timeintervalnode. agg_set_default=Last

```

timeintervalnode 属性	数据类型	属性说明
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	指定是要连续标记记录还是要根据指定日期、时间戳或时间字段构建序列。
field	field	当根据数据构建序列时，指定表示每个记录的日期或时间的字段。
period_start	integer	指定周期或循环周期的起始区间
cycle_start	integer	循环周期的起始周期。
year_start	integer	对于适用的区间类型，指第一个区间所属的年份。
quarter_start	integer	对于适用的区间类型，指第一个区间所属的季度。
month_start	一月 二月 三月 四月 五月 六月 七月 八月 九月 十月 十一月 十二月	
day_start	integer	
hour_start	integer	
minute_start	integer	
second_start	integer	

timeintervalsnode 属性	数据类型	属性说明
periods_per_cycle	integer	对于循环周期，指每个周期中的期间数。
fiscal_year_begins	一月 二月 三月 四月 五月 六月 七月 八月 九月 十月 十一月 十二月	对于季度区间，指定财政年度开始的月份。
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	对于周期性区间（一周中的天、一天中的小时、一天中的分钟和一天中的秒），指定一周开始的那一天。
day_begins_hour	integer	对于周期性区间（一天中的小时、一天中的分钟和一天中的秒），指定一天开始的小时。可以与 day_begins_minute 和 day_begins_second 结合起来使用，以指定一个准确时间，例如 8:05:01。请参阅下面的使用示例。
day_begins_minute	integer	对于周期性区间（一天中的小时、一天中的分钟和一天中的秒），指定一天开始的分钟（例如 8:05 中的 5）。
day_begins_second	integer	对于周期性区间（一天中的小时、一天中的分钟和一天中的秒），指定一天开始的秒（例如 8:05:17 中的 17）。
days_per_week	integer	对于周期性区间（一周中的天、一天中的小时、一天中的分钟和一天中的秒），指定一周中的天数。
hours_per_day	integer	对于周期性区间（一天中的小时、一天中的分钟和一天中的秒），指定一天中的小时数。
interval_increment	1 2 3 4 5 6 10 15 20 30	对于一天中的分钟和一天中的秒，指定为每个记录增加的分钟数或秒数。
field_name_extension	string	

timeintervalnode 属性	数据类型	属性说明
field_name_extension_as_prefix	flag	
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	
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"	
aggregate	Mean Sum Mode Min Max First Last TrueIfAnyTrue	指定字段的汇总方法（例如， aggregate.AGE=Mean）。

timeintervalsnode 属性	数据类型	属性说明
pad	Blank MeanOfRecentPoints True False	指定字段的填充方法（例如，pad. AGE=MeanOfRecentPoints）。
agg_mode	All Specify	指定是根据需要默认使用函数汇总或填充所有字段，还是指定要使用的字段和函数。
agg_range_default	Mean Sum Mode Min Max	指定汇总连续字段时要使用的默认函数。
agg_set_default	Mode First Last	指定汇总名义字段时要使用的默认函数。
agg_flag_default	TrueIfAnyTrue Mode First Last	
pad_range_default	Blank MeanOfRecentPoints	指定填充连续字段时要使用的默认函数。
pad_set_default	Blank MostRecentValue	
pad_flag_default	Blank True False	
max_records_to_create	integer	指定填充序列时要创建的最大记录数。
estimation_from_beginning	flag	
estimation_to_end	flag	
estimation_start_offset	integer	
estimation_num_holdouts	integer	
create_future_records	flag	
num_future_records	integer	
create_future_field	flag	
future_field_name	string	

transposenode 属性



转置节点交换行和列中的数据，以便记录变成字段，字段变成记录。有关详细信息，请参阅第 4 章中的转置节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create transposenode
set :transposenode.transposed_names=Read
set :transposenode.read_from_field="TimeLabel"
set :transposenode.max_num_fields="1000"
```

```
set :transposenode.id_field_name="ID"
```

transposenode 属性	数据类型	属性说明
transposed_names	Prefix Read	可以根据指定前缀自动生成新字段名，也可以从数据的现有字段中读取新字段名。
prefix	string	
num_new_fields	integer	使用前缀时，指定要创建的新字段的最大数目。
read_from_field	field	从中读取名称的字段。此字段必须是一个实例化字段，否则执行节点时将出错。
max_num_fields	integer	当从某个字段中读取名称时，指定上限以避免创建过大的字段数。
transpose_type	Numeric String Custom	默认情况下，只能转置连续（数值范围）字段，但也可以选择数值字段的自定义子集或转置所有字符串字段。
transpose_fields	[field field field]	指定使用 Custom 选项时转置的字段。
id_field_name	field	

typenode 属性



类型节点指定字段元数据和属性。例如，您可以指定每个字段的测量级别（连续、名义、有序或标志）、设置用于处理缺失值和系统空值的选项、设置用于建模的字段的角色、指定字段和值标签，以及为字段指定值。有关详细信息，请参阅第 4 章中的类型节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

注意，某些情况下可能需要完全实例化类型节点才能使其他节点正常运行，例如，设为标志节点的 `fields from` 属性。可以只连接“表”节点并执行该节点以实例化这些字段：

```
create tablenode
connect :typenode to :tablenode
```

execute :tablenode

delete :tablenode

typenode 属性	数据类型	属性说明
direction	Input Target Both None Partition Split Frequency RecordID	字段角色的键控属性。 用法格式： NODE.direction.FIELDNAME 注意：值 In 和 Out 现已无效。在将来的版本中可能取消对这些值的支持。
type	Range Flag Set Typeless Discrete Ordered Set Default	字段类型。将 type 设置为 Default 会清除所有的 values 参数设置，如果 value_mode 的值为 Specify，则其值将被重置为 Read。如果将 value_mode 设置为 Pass 或 Read，则设置 type 将不会影响 value_mode。 用法格式： NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	字段存储类型的只读键控属性。 用法格式： NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	字段类型和范围检查的键控属性。 用法格式： NODE.check.FIELDNAME
values	[value value]	对于连续字段而言，第一个是最小值，后一个是最大值。对于名义字段，指定所有值。对标志字段而言，第一个值代表 false，后一个值代表 true。设置该属性将自动把 value_mode 属性设置为 Specify。 用法格式： NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	确定值的设置方式。注意，不能将此属性直接设置为 Specify；要使用特定值，需设置 values 属性。 用法格式： NODE.value_mode.FIELDNAME
extend_values	flag	当 value_mode 设置为 Read 时可应用。设为 T 则将新读取的值添加到任意现有字段值。设置为 F 则丢弃现有值并添加新读取值。 用法格式： NODE.extend_values.FIELDNAME
enable_missing	flag	当设置为 T 时，则激活对字段缺失值的跟踪。 用法格式： NODE.enable_missing.FIELDNAME

typenode 属性	数据类型	属性说明
missing_values	[value value ...]	指定表示缺失数据的数据值。 用法格式： NODE.missing_values.FIELDNAME
range_missing	flag	指定是否为字段定义缺失值（空白）范围。
missing_lower	string	当 range_missing 为真时，指定缺失值范围的下限。
missing_upper	string	当 range_missing 为真时，指定缺失值范围的上限。
null_missing	flag	设置为 T 时，nulls（在软件中显示为 \$null\$ 的未定义值）被视为缺失值。 用法格式： NODE.null_missing.FIELDNAME
whitespace_missing	flag	设置为 T 时，仅包含空白（空格、制表符和换行符）的值被视为缺失值。 用法格式： NODE.whitespace_missing.FIELDNAME
description	string	为字段指定说明。
value_labels	[[{Value LabelString} { Value LabelString} ...]]	用于为值对指定标签。 示例如下： set :typenode.value_labels.'Drug'=[{drugA label1} {drugB label2}]
display_places	integer	为字段设置显示的小数位数（仅用于以 REAL 存储的字段）。值为 -1 时将使用流默认值。 用法格式： NODE.display_places.FIELDNAME
export_places	integer	为字段设置导出时的小数位数（仅用于以 REAL 存储的字段）。值为 -1 时将使用流默认值。 用法格式： NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD COMMA	为字段设置十进制分隔符（仅用于以 REAL 存储的字段）。 用法格式： NODE.decimal_separator.FIELDNAME
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"	为字段设置日期格式（仅用于以 DATE 或 TIMESTAMP 存储的字段）。 用法格式： NODE.date_format.FIELDNAME 示例如下： set :tablenode.date_format.'LaunchDate' = "DDMMYY"

typenode 属性	数据类型	属性说明
	"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	
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"	为字段设置时间格式（仅用于以 TIME 或 TIMESTAMP 存储的字段）。 用法格式： NODE.time_format.FIELDNAME 示例如下： set :tablnode.time_format.'BOF_enter' = "HHMMSS"
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	为字段设置数字显示格式。 用法格式： NODE.number_format.FIELDNAME
standard_places	integer	为字段设置以标准格式显示时的小数位数。值为 -1 时将使用流默认值。 请注意，现有的 display_places 通道也会更改此属性，但目前已不再使用。 用法格式： NODE.standard_places.FIELDNAME
scientific_places	integer	为字段设置以科学计数格式显示时的小数位数。值为 -1 时将使用流默认值。 用法格式： NODE.scientific_places.FIELDNAME
currency_places	integer	为字段设置以货币格式显示时的小数位数。值为 -1 时将使用流默认值。 用法格式： NODE.currency_places.FIELDNAME
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	为字段设置分组符号。 用法格式： NODE.grouping_symbol.FIELDNAME

typenode 属性	数据类型	属性说明
column_width	integer	为字段设置列宽度。值为 -1 表示将列宽度设置为 Auto。 用法格式： NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	为字段设置列对齐格式。 用法格式： NODE.justify.FIELDNAME

图形节点属性

图形节点通用属性

本节介绍图形节点的可用属性，包括通用属性和每种节点类型特有的属性。

通用图形节点属性	数据类型	属性说明
title	string	指定标题。示例：” This is a title.”
caption	string	指定标注。示例：” This is a caption.”
output_mode	Screen File	指定是显示图形节点的输出还是将其写到文件中。
output_format	BMP JPEG PNG HTML output (.cou)	指定输出的类型。允许每个节点使用的确切输出类型是不同的。
full_filename	string	为从图形节点生成的输出指定目标路径和文件名。
use_graph_size	flag	控制是否使用下面的宽度和高度属性明确调整图形的大小。只影响输出到屏幕的图形。不适用于分布节点。
graph_width	number	当 use_graph_size 为 True 时，以像素为单位设置图形宽度。
graph_height	number	当 use_graph_size 为 True 时，以像素为单位设置图形高度。

附注

关闭可选字段。 通过将属性值设置为 ” ”（空字符串）可将可选字段（如图的重叠字段）关闭，如以下示例所示：

```
set :plotnode.color_field = ""
```

指定颜色。 使用十六进制字符串（从井号 (#) 开始），可指定标题、标注、背景和标签的颜色。例如，要将图形背景设置为天蓝色，请使用以下语句：

```
set mygraph.graph_background="#87CEEB"
```

此处，前两位数 87 指定红色内容；中间两位数 CE 指定绿色内容；最后两位数 EB 指定蓝色内容。每位数可获取一个位于范围 0-9 或 A-F 内的值。这些值在一起可以指定红-绿-蓝（即 RGB）颜色。

注意：以 RGB 形式指定颜色时，可以使用用户界面中的字段选择器确定正确的颜色代码。只需将鼠标停留在颜色上面就可激活含所需信息的工具提示。

collectionnode 属性



“收集”节点显示一个数字字段的值相对于另一个数字字段的值的分布。（它创建类似于直方图的图形。）图示说明值不断变化的变量或字段时，它是有用的。使用 3-D 图形表示时，还可以使用按分类显示分布的符号轴。有关详细信息，请参阅第 5 章中的收集散点图选项卡中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

collectionnode 属性	数据类型	属性说明
over_field	field	
over_label_auto	flag	
over_label	string	
collect_field	field	
collect_label_auto	flag	
collect_label	string	
three_D	flag	
by_field	field	
by_label_auto	flag	
by_label	string	
operation	Sum Mean Min Max SDev	
color_field	string	
panel_field	string	
animation_field	string	

collectionnode 属性	数据类型	属性说明
range_mode	Automatic UserDefined	
range_min	number	
range_max	number	
bins	ByNumber ByWidth	
num_bins	number	
bin_width	number	
use_grid	flag	
graph_background	颜色	本节在开头处介绍了标准图形颜色。
page_background	颜色	本节在开头处介绍了标准图形颜色。

distributionnode 属性



条形图节点显示了标志（类别）值的出现次数，例如抵押类型或性别。通常可以使用条形图结点来显示数据中的不均衡，然后可在模型创建前使用均衡节点来纠正此类不均衡。有关详细信息，请参阅第 5 章中的条形图节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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 = ByOccurence
set :distributionnode.use_proportional_scale = True
```

distributionnode 属性	数据类型	属性说明
plot	SelectedFields Flags	
x_field	field	
color_field	field	交叠字段。
normalize	flag	
sort_mode	ByOccurence Alphabetic	
use_proportional_scale	flag	

evaluationnode 属性



评估节点有助于评估和比较预测模型。评估图表显示了模型对特定结果的预测优劣。它根据预测值和预测置信度来对记录进行排序。它将记录分成若干个相同大小的组（分位数），然后从高到底为每个分位数划分业务标准值。在散点图中，将以单独的线条显示多个模型。有关详细信息，请参阅第 5 章中的评估节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

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'

```

evaluationnode 属性	数据类型	属性说明
chart_type	Gains Response Lift Profit ROI	
inc_baseline	flag	
field_detection_method	Metadata Name	
use_fixed_cost	flag	
cost_value	number	
cost_field	string	
use_fixed_revenue	flag	
revenue_value	number	
revenue_field	string	
use_fixed_weight	flag	
weight_value	number	
weight_field	field	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	flag	
style	Line Point	

evaluationnode 属性	数据类型	属性说明
point_type	矩形 点 三角形 六角形 加 五角形 星形 BowTie HorizontalDash VerticalDash IronCross 工厂 房子 教堂 OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle 扇形	
export_data	flag	
data_filename	string	
delimiter	string	
new_line	flag	
inc_field_names	flag	
inc_best_line	flag	
inc_business_rule	flag	
business_rule_condition	string	
plot_score_fields	flag	
score_fields	[field1 ... fieldN]	
target_field	field	
use_hit_condition	flag	
hit_condition	string	
use_score_expression	flag	
score_expression	string	
caption_auto	flag	

graphboardnode 属性



图形板节点可在一个节点中提供许多不同类型的图形。使用此节点，可以选择要探索的数据字段，然后从适用于选定数据的字段中选择一个图形。节点将自动过滤出适用于字段选项的所有图形类型。有关详细信息，请参阅第 5 章中的图形板节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

注意：如果您设置对图形类型无效的属性（例如，为直方图指定 `y_field`），该属性将被忽略。

示例

```

create graphboardnode
connect DRUG4n to :graphboardnode
set :graphboardnode.graph_type="Line"
set :graphboardnode.x_field = "K"
set :graphboardnode.y_field = "Na"
execute :graphboardnode

```

graphboard 属性	数据类型	属性说明
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 CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap PointsOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	标识图形类型。
x_field	field	为 x 轴指定自定义标签。只适用于标签。

graphboard 属性	数据类型	属性说明
y_field	field	为 y 轴指定自定义标签。只适用于标签。
z_field	field	在某些 3D 图形中使用。
color_field	field	在热图中使用。
size_field	field	在气泡散点图中使用。
categories_field	field	
values_field	field	
rows_field	field	
columns_field	field	
fields	field	
start_longitude	field	用于参考地图上的箭头。
end_longitude	field	
start_latitude	field	
end_latitude	field	
data_key_field	field	在多种地图中使用。
panelrow_field	string	
panelcol_field	string	
animation_field	string	
longitude	field	用于地图上的坐标。
latitude	field	
map_color_field	field	

histogramnode 属性



直方图节点显示了数值字段的值的出现次数。它经常用来在数据操作和模型构建之前探索数据。与条形图节点相似，直方图节点经常用来揭示数据中的不均衡。有关详细信息，请参阅第 5 章中的直方图选项卡中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

histogramnode 属性	数据类型	属性说明
field	field	
color_field	field	
panel_field	field	
animation_field	field	
range_mode	Automatic UserDefined	
range_min	number	
range_max	number	
bins	ByNumber ByWidth	
num_bins	number	
bin_width	number	
normalize	flag	
separate_bands	flag	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
use_grid	flag	
graph_background	颜色	本节在开头处介绍了标准图形颜色。
page_background	颜色	本节在开头处介绍了标准图形颜色。
normal_curve	flag	指出是否应在输出中显示正态分布曲线。

multiplotnode 属性



使用多重散点图节点可创建一个 X 字段上显示多个 Y 字段的散点图。Y 字段被绘制为彩色的线；每条线相当于“样式”设置为线且“X 模式”设置为排序的散点图节点。当要研究几个变量随时间的变化情况时，多重散点图非常有用。有关详细信息，请参阅第 5 章中的多重散点图节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

multiplotnode 属性	数据类型	属性说明
x_field	field	
y_fields	[field field field]	
panel_field	field	
animation_field	field	
normalize	flag	
use_overlay_expr	flag	
overlay_expression	string	
records_limit	number	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
use_grid	flag	
graph_background	颜色	本节在开头处介绍了标准图形颜色。
page_background	颜色	本节在开头处介绍了标准图形颜色。

plotnode 属性



散点图节点可显示数值字段间的关系。可通过使用点（散点）或线创建散点图。有关详细信息，请参阅第 5 章中的散点图节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

plotnode 属性	数据类型	属性说明
x_field	field	为 x 轴指定自定义标签。只适用于标签。
y_field	field	为 y 轴指定自定义标签。只适用于标签。
three_D	flag	为 y 轴指定自定义标签。只适用于 3 维图形中的标签。
z_field	field	
color_field	field	交叠字段。
size_field	field	
shape_field	field	
panel_field	field	指定用于为每个类别绘制单独图表的名义字段或标志字段。图表一起平铺在一个输出窗口中。
animation_field	field	指定以图说明数据值类别（通过使用动画创建一系列按顺序显示的图表来说明）时所使用的名义字段或标志字段。
transp_field	field	指定以图说明数据值类别（通过为每个类别使用不同级别的透明度来说明）时所使用的字段。不适用于线条图。
overlay_type	None Smoother Function	指定是显示重叠函数还是 LOESS 平滑器。
overlay_expression	string	指定当 overlay_type 设置为 Function 时使用的表达式。
style	Point Line	
point_type	矩形 点 三角形 六角形 加 五角形 星形 BowTie HorizontalDash VerticalDash IronCross 工厂 房子 教堂 OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle 扇形	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	

plotnode 属性	数据类型	属性说明
x_range_min	number	
x_range_max	number	
y_range_mode	Automatic UserDefined	
y_range_min	number	
y_range_max	number	
z_range_mode	Automatic UserDefined	
z_range_min	number	
z_range_max	number	
jitter	flag	
records_limit	number	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
z_label_auto	flag	
z_label	string	
use_grid	flag	
graph_background	颜色	本节在开头处介绍了标准图形颜色。
page_background	颜色	本节在开头处介绍了标准图形颜色。
use_overlay_expr	flag	该属性已由 <code>overlay_type</code> 替代。

timeplotnode 属性



时间散点图节点显示一个或多个时间序列数据集。通常情况下，您首先要使用时间区间节点创建一个 TimeLabel 字段，该字段用于为 x 轴设置标签。有关详细信息，请参阅第 5 章中的时间散点图节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

timeplotnode 属性	数据类型	属性说明
plot_series	Series Models	
use_custom_x_field	flag	
x_field	field	
y_fields	[field field field]	
panel	flag	
normalize	flag	
line	flag	
points	flag	
point_type	矩形 点 三角形 六角形 加 五角形 星形 BowTie HorizontalDash VerticalDash IronCross 工厂 房子 教堂 OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle 扇形	
smoother	flag	只有将 <code>panel</code> 设置为 <code>True</code> ，才可将平滑器添加到图中。
use_records_limit	flag	
records_limit	integer	
symbol_size	number	指定符号大小。 例如， <code>set :webnode.symbol_size = 5.5</code> 可创建大于预设的符号大小。
panel_layout	Horizontal Vertical	

webnode 属性



Web 节点说明了两个或多个符号（分类）字段值之间关系的强度。该图使用不同粗细的线来表示关系强度。例如，您可以使用 Web 节点来研究电子商务网站上系列商品的购买之间的关系。有关详细信息，请参阅第 5 章中的网络节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

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

```

webnode 属性	数据类型	属性说明
use_directed_web	flag	
fields	[field field field]	
to_field	field	
from_fields	[field field field]	
true_flags_only	flag	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	flag	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	number	
links_above	number	
discard_links_min	flag	
links_min_records	number	
discard_links_max	flag	
links_max_records	number	
weak_below	number	
strong_above	number	
link_size_continuous	flag	

webnode 属性	数据类型	属性说明
web_display	Circular Network Directed Grid	
graph_background	颜色	本节在开头处介绍了标准图形颜色。
symbol_size	number	指定符号大小。 例如， <code>set :webnode.symbol_size = 5.5</code> 可创建大于预设的符号大小。

建模节点属性

通用建模节点属性

以下属性通用于某些或所有建模节点。所有例外情况均根据需要记录在各个建模节点的文档中。

属性	值	属性说明
custom_fields	flag	如果为 true, 则允许您指定当前节点的目标、输入和其他字段。如果为 false, 则使用来自上游类型节点的当前设置。
target 或 targets	field 或 [field1 ... fieldN]	根据模型类型, 指定单个目标字段或多个目标字段。
inputs	[field1 ... fieldN]	模型使用的输入或预测变量字段。
partition	field	
use_partitioned_data	flag	如果定义了分区字段, 则此选项可确保仅训练分区的数据用于构建模型。
use_split_data	flag	
splits	[field1 ... fieldN]	指定拆分建模使用的字段。仅当 use_split_data 设置为 True 时有效。
use_frequency	flag	正如每种模型类型的注释, 某些特定模型会使用加权或频率字段。
frequency_field	field	
use_weight	flag	
weight_field	field	
use_model_name	flag	
model_name	string	新模型的自定义名称。
mode	Simple Expert	

anomalydetectionnode 属性



“异常检测”节点确定不符合“正常”数据格式的异常观测值（离群值）。即使离群值不匹配任何已知格式或用户不清楚自己的查找对象，也可以使用此节点来确定离群值。有关详细信息，请参阅第 4 章中的异常检测节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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

```

anomalydetectionnode 属性	值	属性说明
inputs	[field1 ... fieldN]	异常检测模型屏幕根据指定的输入字段进行记录。它们不使用目标字段。也不使用加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	指定用于确定将记录标记为异常的截断值。
index_level	number	指定标记异常的最小截断值。
percent_records	number	根据训练数据中记录的百分比设置标记记录的阈值。
num_records	number	根据训练数据中的记录数设置标记记录的阈值。
num_fields	integer	每条异常记录报告的字段数。
impute_missing_values	flag	
adjustment_coeff	number	用于平衡在计算距离时赋予连续和分类字段的相对权重的数值。
peer_group_num_auto	flag	自动计算对等组数。
min_num_peer_groups	integer	指定 peer_group_num_auto 设置为 True 时使用的最小对等组数。
max_num_per_groups	integer	指定最大对等组数。
num_peer_groups	integer	指定 peer_group_num_auto 设置为 False 时使用的对等组数。
noise_level	number	确定创建聚类期间离群值的处理方式。请指定介于 0 和 0.5 之间的值。
noise_ratio	number	指定分配给用于噪声缓冲的组件的内存量。请指定介于 0 和 0.5 之间的值。

apriorinode 属性



Apriori 节点从数据抽取一组规则，即抽取信息内容最多的规则。Apriori 节点提供五种选择规则的方法并使用复杂的索引模式来高效地处理大数据集。对于较大的问题，Apriori 训练的速度通常较快；它对可保留的规则数量没有任何限制，而且可处理最多带有 32 个前提条件的规则。Apriori 要求输入和输出字段均为分类型字段，但因为它专为处理此类型数据而进行优化，因而处理速度快得多。有关详细信息，请参阅第 12 章中的 Apriori 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```

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'
# These seem to have changed, used to be:
#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

```

apriorinode 属性	值	属性说明
consequents	field	Apriori 模型使用“结果”和“条件”代替标准目标和输入字段。也不使用加权和频率字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
antecedents	[field1 ... fieldN]	
min_supp	number	
min_conf	number	
max_antecedents	number	
true_flags	flag	
optimize	Speed Memory	
use_transactional_data	flag	
contiguous	flag	
id_field	string	
content_field	string	
mode	Simple Expert	

apriorinode 属性	值	属性说明
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	number	
optimize	Speed Memory	用于指定优化建模的方式是速度还是内存。

autoclassifiernode 属性



“自动分类器”节点用于创建和对比二元结果（是或否，流失或不流失等）的若干不同模型，使用户可以选择给定分析的最佳处理方法。由于支持多种建模算法，因此可以对用户希望使用的方法、每种方法的特定选项以及对比结果的标准进行选择。节点根据指定的选项生成一组模型并根据用户指定的标准排列最佳候选项的顺序。有关详细信息，请参阅第 5 章中的自动分类器节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create autoclassifiernode
set :autoclassifiernode.ranking_measure=Accuracy
set :autoclassifiernode.ranking_dataset=Training
set :autoclassifiernode.enable_accuracy_limit=true
set :autoclassifiernode.accuracy_limit=0.9
set :autoclassifiernode.calculate_variable_importance=true
set :autoclassifiernode.use_costs=true
set :autoclassifiernode.svm=false
```

autoclassifiernode 属性	值	属性说明
target	field	对于标志目标，自动分类器节点要求单个目标字段以及一个或多个输入字段。还可以指定加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	integer	要在模型块中包括的模型数量。指定整数必须为 1 到 100 之间。
calculate_variable_importance	flag	
enable_accuracy_limit	flag	
accuracy_limit	integer	0 到 100 之间的整数。
enable_area_under_curve_limit	flag	

autoclassifiernode 属性	值	属性说明
area_under_curve_limit	number	0.0 到 1.0 之间的实数。
enable_profit_limit	flag	
profit_limit	number	大于 0 的整数。
enable_lift_limit	flag	
lift_limit	number	大于 1.0 的实数。
enable_number_of_variables_limit	flag	
number_of_variables_limit	number	大于 0 的整数。
use_fixed_cost	flag	
fixed_cost	number	大于 .0 的实数。
variable_cost	field	
use_fixed_revenue	flag	
fixed_revenue	number	大于 .0 的实数。
variable_revenue	field	
use_fixed_weight	flag	
fixed_weight	number	大于 0.0 的实数。
variable_weight	field	
lift_percentile	number	0 到 100 之间的整数。
enable_model_build_time_limit	flag	
model_build_time_limit	number	设置为分钟数的整数，用于限制构建各个模型所花费的时间。
enable_stop_after_time_limit	flag	
stop_after_time_limit	number	设置为小时数的实数集，用于限制运行自动分类器的总耗时。
enable_stop_after_valid_model_produced	flag	
use_costs	flag	
<algorithm>	flag	启用或禁用特定算法，例如： set :autoclassifiernode.chaid=true
<algorithm>. <property>	string	为特定算法设置属性值。 有关详细信息，请参阅第 175 页码设置算法属性。

设置算法属性

对于自动分类器、自动数值和自动聚类节点，可以采用一般形式设置节点所使用的特定算法的属性：

```
set :autoclassifiernode.<algorithm>.<property> = <value>
```

```
set :autonumericnode.<algorithm>.<property> = <value>
```

```
set :autoclusternode.<algorithm>.<property> = <value>
```

例如：

```
set :autoclassifiernode.neuralnetwork.method = MultilayerPerceptron
```

用于自动分类器节点的算法名称有 `cart`、`chaid`、`quest`、`c50`、`logreg`、`decisionlist`、`bayesnet`、`discriminant`、`svm` 和 `knn`。

用于自动数值节点的算法名称有 `cart`、`chaid`、`neuralnetwork`、`genlin`、`svm`、`regression`、`linear` 和 `knn`。

用于自动聚类节点的算法名称有 `twostep`、`k-means` 和 `kohonen`。

属性名为各节点文档中记录的标准格式。

含有句点和其他标点符号的算法属性必须包括在半角单引号中，例如：

```
set :autoclassifiernode.logreg.tolerance = '1.0E-5'
```

也可以为属性分配多个值，例如：

```
set :autoclassifiernode.decisionlist.search_direction = [Up Down]
```

要启用或禁用特定算法：

```
set :autoclassifiernode.chaid=true
```

注意：

- 设置 `true` 和 `false` 值时，必须使用小写字符（而不是 `False`）。
- 如果自动分类器节点中某些算法选项不可用，或只能指定单个值，而不是某个范围的值，则编写脚本时受到的限制与以标准方式访问节点时一样。

autoclusternode 属性



自动聚类节点估算和比较识别具有类似特征记录组的聚类模型。节点工作方式与其他自动建模节点相同，使您在一次建模运行中即可试验多个选项组合。模型可使用基本测量进行比较，以尝试过滤聚类模型的有效性以及对其进行排序，并提供一个基于特定字段的重要性的测量。[有关详细信息，请参阅第 5 章中的自动聚类节点中的 IBM SPSS Modeler 15 建模节点。](#)

示例

```
create autoclusternode
set :autoclusternode.ranking_measure=Silhouette
set :autoclusternode.ranking_dataset=Training
set :autoclusternode.enable_silhouette_limit=true
```

```
set :autoclusternode.silhouette_limit=5
```

autoclusternode 属性	值	属性说明
evaluation	field	注意：仅自动聚类节点。标识将为其计算重要性值的字段。此外，还可用于确定聚类区分此字段值的程度，从而确定模型预测此字段的程度。
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	integer	要列明在报告中的模型数量。指定整数必须为 1 到 100 之间。
enable_silhouette_limit	flag	
silhouette_limit	integer	0 到 100 之间的整数。
enable_number_less_limit	flag	
number_less_limit	number	0.0 到 1.0 之间的实数。
enable_number_greater_limit	flag	
number_greater_limit	number	大于 0 的整数。
enable_smallest_cluster_limit	flag	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	number	
smallest_cluster_limit_count	integer	大于 0 的整数。
enable_largest_cluster_limit	flag	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percentage	number	
largest_cluster_limit_count	integer	
enable_smallest_largest_limit	flag	
smallest_largest_limit	number	
enable_importance_limit	flag	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	number	0 到 100 之间的整数。
importance_limit_less_than	number	0 到 100 之间的整数。

autoclusternode 属性	值	属性说明
<algorithm>	flag	启用或禁用特定算法，例如： set :autoclusternode.kohonen=true
<algorithm>. <property>	string	为特定算法设置属性值。 有关详细信息，请参阅第 175 页码设置算法属性。

autonumericnode 属性



自动数值节点使用多种不同方法估计和对比模型的连续数字范围结果。此节点和自动分类器节点的工作方式相同，因此可以选择要使用和要在单个建模传递中使用多个选项组合进行测试的算法。受支持的算法包括神经网络、C&R 树、CHAID、线性回归、广义线性回归以及 Support Vector Machine (SVM)。可基于相关度、相对错误或已用变量数对模型进行对比。 [有关详细信息，请参阅第 5 章中的自动数值节点中的 IBM SPSS Modeler 15 建模节点。](#)

示例

```
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
```

autonumericnode 属性	值	属性说明
custom_fields	flag	如果为 True，则将使用自定义字段设置来代替类型节点设置。
target	field	自动数值节点要求单个目标字段以及一个或多个输入字段。还可以指定加权和频率字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
inputs	[field1 ... field2]	
partition	field	
use_frequency	flag	
frequency_field	field	
use_weight	flag	
weight_field	field	
use_partitioned_data	flag	如果定义了分区字段，则建模时仅使用训练数据。
ranking_measure	Correlation NumberOfFields	
ranking_dataset	Test Training	
number_of_models	integer	要在模型块中包括的模型数量。指定整数必须为 1 到 100 之间。

autonumericnode 属性	值	属性说明
calculate_variable_importance	flag	
enable_correlation_limit	flag	
correlation_limit	integer	
enable_number_of_fields_limit	flag	
number_of_fields_limit	integer	
enable_relative_error_limit	flag	
relative_error_limit	integer	
enable_model_build_time_limit	flag	
model_build_time_limit	integer	
enable_stop_after_time_limit	flag	
stop_after_time_limit	integer	
stop_if_valid_model	flag	
<algorithm>	flag	启用或禁用特定算法，例如： set :autonumericnode. chaid=true
<algorithm>. <property>	string	为特定算法设置属性值。 有关详细信息，请参阅第 175 页码设置算法属性。

bayesnetnode 属性



通过贝叶斯网络节点，你可以利用对真实世界认知的判断力并结合所观察和记录的证据来构建概率模型。该节点重点应用了树扩展简单贝叶斯 (TAN) 和马尔可夫毯网络，这些算法主要用于分类问题。 [有关详细信息，请参阅第 7 章中的贝叶斯网络节点中的 IBM SPSS Modeler 15 建模节点。](#)

示例

```
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
```

bayesnetnode 属性	值	属性说明
inputs	[field1 ... fieldN]	Bayesian 网络模型使用单个目标字段，以及一个或多个输入字段。连续字段将自动分隔。 有关详细信息，请参阅第 171 页码通用建模节点属性。
continue_training_existing_model	flag	

bayesnetnode 属性	值	属性说明
structure_type	TAN MarkovBlanket	选择构建贝叶斯网络时使用的结构。
use_feature_selection	flag	
parameter_learning_method	Likelihood Bayes	指定用于预测父节点的值已知的节点之间的条件概率表的方法。
mode	Expert Simple	
missing_values	flag	
all_probabilities	flag	
independence	Likelihood Pearson	指定用于决定两个变量中的成对观测值是否相互独立的方法。
significance_level	number	指定用于确定独立性的截断值。
maximal_conditioning_set	number	设置执行独立性检验时要使用的条件变量的最大数目。
inputs_always_selected	[field1 ... fieldN]	指定构建贝叶斯网络时始终使用的数据集中的字段。 注意：始终选择目标字段。
maximum_number_inputs	number	指定了构建贝叶斯网络时使用的输入字段的最大数目。
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

c50node 属性



C5.0 节点构建决策树或规则集。该模型的工作原理是根据在每个级别提供最大信息收获的字段分割样本。目标字段必须为分类字段。允许进行多次多于两个子组的分割。有关详细信息，请参阅第 6 章中的 C5.0 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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}]
```

c50node 属性	值	属性说明
target	field	C50 模型使用单个目标字段，以及一个或多个输入字段。还可以指定加权字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
output_type	DecisionTree RuleSet	
group_symbolics	flag	
use_boost	flag	
boost_num_trials	number	
use_xval	flag	
xval_num_folds	number	
mode	Simple Expert	
favor	Accuracy Generality	首选准确性或通用性。
expected_noise	number	
min_child_records	number	
pruning_severity	number	
use_costs	flag	
costs	structured	这是结构化属性。
use_winning	flag	
use_global_pruning	flag	默认为“启用”(True)。
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

carmanode 属性



CARMA 模型在不要求用户指定输入或目标字段的情况下从数据抽取一组规则。与 Apriori 不同的是：CARMA 节点提供构建规则设置支持（前项和后项支持），而不仅仅是前项支持。这就意味着生成的规则可以用于更多应用程序，例如用于找到后项为想在节日期间促销的商品的产品或服务（前项）的列表。有关详细信息，请参阅第 12 章中的 CARMA 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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

```

carmanode 属性	值	属性说明
inputs	[field1 ... fieldn]	CARMA 模型使用一张输入字段列表, 但没有目标字段。也不使用加权和频率字段。有关详细信息, 请参阅第 171 页码通用建模节点属性。
id_field	field	用作模型构建 ID 字段的字段。
contiguous	flag	用于指定 ID 字段中的 ID 是否连续。
use_transactional_data	flag	
content_field	field	
min_supp	number(percent)	与规则支持相关, 而不是与条件支持相关。默认值为 20%。
min_conf	number(percent)	默认值为 20%。
max_size	number	默认值为 10。
mode	Simple Expert	默认值为 Simple。
exclude_multiple	flag	排除具有多结果的规则。默认值为 False。
use_pruning	flag	默认值为 False。
pruning_value	number	默认值为 500。
vary_support	flag	
estimated_transactions	integer	
rules_without_antecedents	flag	

cartnode 属性



分类和回归 (C&R) 树节点生成可用于预测或分类未来观测值的决策树。该方法通过在每个步骤最大限度降低不纯度，使用递归分区来将训练记录分割为组。如果节点中 100% 的观测值都属于目标字段的一个特定类别，则树中的该节点将被认定为“纯洁”。目标和输入字段可以是数字范围或分类（名义、有序或标志）；所有分割均为二元分割（即仅分割为两个子组）。有关详细信息，请参阅第 6 章中的 C&R 树节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
# custom priors must add to 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"
```

cartnode 属性	值	属性说明
target	field	C&R 树模型要求单个目标字段，以及一个或多个输入字段。还可以指定频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
continue_training_existing_model	flag	

cartnode 属性	值	属性说明
objective	Standard Boosting Bagging psm	psm 用于非常大的数据集，同时需要 Server 连接。
model_output_type	Single InteractiveBuilder	
use_tree_directives	flag	
tree_directives	string	指定生成树的指令。指令可放在三重引号（"""）中，以防止丢失新行或引号。请注意，指令可能对数据或建模选项的细小更改高度敏感，并且可能无法通用于其他数据集。有关详细信息，请参阅第 6 章中的树生长指令中的 IBM SPSS Modeler 15 建模节点。
use_max_depth	Default Custom	
max_depth	integer	最大树深度从 0 到 1000。只在 use_max_depth = Custom 时使用。
prune_tree	flag	修剪树以防止过拟合。
use_std_err	flag	使用最大风险差值（标准误）。
std_err_multiplier	number	最大差值。
max_surrogates	number	最大代用项。
use_percentage	flag	
min_parent_records_pc	number	
min_child_records_pc	number	
min_parent_records_abs	number	
min_child_records_abs	number	
use_costs	flag	
costs	structured	使用以下格式的结构化属性： [{drugA drugB 1.5} {drugA drugC 2.1}]，其中花括号（{}）中的参数为实际预测成本。
priors	Data Equal Custom	
custom_priors	structured	使用以下格式的结构化属性： set :cartnode. custom_priors = [{ drugA 0.3 } { drugB 0.6 }]
adjust_priors	flag	
trails	number	用于推进或 bagging 的成分模型数。
set_ensemble_method	Voting HighestProbability HighestMeanProbability	分类目标的默认组合规则。
range_ensemble_method	Mean Median	连续目标的默认组合规则。
large_boost	flag	将推进应用到大型数据集。

cartnode 属性	值	属性说明
min_impurity	number	
impurity_measure	Gini Twoing Ordered	
train_pct	number	防止过度拟合集合。
set_random_seed	flag	重复结果选项。
seed	number	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

chaidnode 属性



CHAID 使用卡方统计量来生成决策树，以确定最佳的分割。CHAID 与 C&R 树和 QUEST 节点不同，它可以生成非二元树，这意味着有些分割将有多于两个的分支。目标和输入字段可以是数字范围（连续）或分类。Exhaustive CHAID 是 CHAID 的修正版，它对所有分割进行更彻底的检查，但计算时间比较长。有关详细信息，请参阅第 6 章中的 CHAID 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

chaidnode 属性	值	属性说明
target	field	CHAID 模型要求使用单个目标字段，以及一个或多个输入字段。还可以指定频率字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
continue_training_existing_model	flag	
objective	Standard Boosting Bagging psm	psm 用于非常大的数据集，同时需要 Server 连接。
model_output_type	Single InteractiveBuilder	
use_tree_directives	flag	
tree_directives	string	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	integer	最大树深度从 0 到 1000。只在 use_max_depth = Custom 时使用。
use_percentage	flag	
min_parent_records_pc	number	
min_child_records_pc	number	
min_parent_records_abs	number	
min_child_records_abs	number	
use_costs	flag	
costs	structured	使用以下格式的结构化属性： [[drugA drugB 1.5] {drugA drugC 2.1}], 其中花括号 ({}) 中的参数为实际预测成本。
trails	number	用于推进或 bagging 的成分模型数。
set_ensemble_method	Voting HighestProbability HighestMeanProbability	分类目标的默认组合规则。
range_ensemble_method	Mean Median	连续目标的默认组合规则。
large_boost	flag	将推进应用到大型数据集。
split_alpha	number	用于分割的显著性水平。
merge_alpha	number	用于合并的显著性水平。
bonferroni_adjustment	flag	使用 Bonferroni 方法调整显著性值。
split_merged_categories	flag	允许重新分割合并的类别。
chi_square	Pearson LR	用于计算卡方统计量的方法：Pearson 或似然比
epsilon	number	期望单元格频率的最小更改。

chaidnode 属性	值	属性说明
max_iterations	number	收敛的最大迭代次数。
set_random_seed	integer	
seed	number	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	integer	

coxregnode 属性



使用 Cox 回归节点，您可以在已有的检查记录中建立时间事件的生存模型。该模型会生成一个生存函数，该函数可预测在给定时间 (t) 内对于所给定的输入变量值相关事件的发生概率。有关详细信息，请参阅第 10 章中的 Cox 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

coxregnode 属性	值	属性说明
survival_time	field	Cox 回归模型要求单个字段包含生存时间。
target	field	Cox 回归模型要求单个目标字段，以及一个或多个输入字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
method	Enter Stepwise BackwardsStepwise	
groups	field	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	示例： set :coxregnode. custom_terms = ["BP*Sex" "BP" "Age"]
mode	Expert Simple	
max_iterations	number	

coxregnode 属性	值	属性说明
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	number	
probability_removal	number	
output_display	EachStep LastStep	
ci_enable	flag	
ci_value	90 95 99	
correlation	flag	
display_baseline	flag	
survival	flag	
hazard	flag	
log_minus_log	flag	
one_minus_survival	flag	
separate_line	field	
value	数字或字符串	如果字段没有指定任何值，则该字段将使用默认选项“Mean（均值）”。 数字字段的使用方法： coxnode.value = [{"age" "35.8"}] 分类字段的使用方法： coxnode.value = [{"color" "pink"}]

decisionlistnode 属性



决策列表节点可标识子组或段，显示与总体相关的给定二元结果的似然度的高低。例如，您或许在寻找那些最不可能流失的客户或最有可能对某个商业活动作出积极响应的客户。通过定制段和并排预览备选模型来比较结果，您可以将自己的业务知识体现在模型中。决策列表模型由一组规则构成，其中每个规则具备一个条件和一个结果。规则依顺序应用，相匹配的第一个规则将决定结果。有关详细信息，请参阅第 9 章中的决策表中的 IBM SPSS Modeler 15 建模节点。

示例

```
create decisionlistnode
set :decisionlistnode.search_direction=Down
set :decisionlistnode.target_value=1
set :decisionlistnode.max_rules=4
set :decisionlistnode.min_group_size_pct = 15
```

decisionlistnode 属性	值	属性说明
target	field	决策列表模型使用单个目标字段，以及一个或多个输入字段。还可以指定频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	与查找段相关；其中“上”相当于高概率，“下”相当于低概率..
target_value	string	如果未指定，则假定为标志的真值。
max_rules	integer	除余数之外的最大段数。
min_group_size	integer	最小段大小。
min_group_size_pct	number	以百分比表示的最小段大小。
confidence_level	number	为了使输入字段符合添加到段定义的条件，输入字段必须提高的响应似然值（提升）的最小阈值。
max_segments_per_rule	integer	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	number	
max_models_per_cycle	integer	列表搜索宽度。
max_rules_per_cycle	integer	段规则的搜索宽度。
segment_growth	number	
include_missing	flag	
final_results_only	flag	
reuse_fields	flag	允许重复使用的特性（出现在规则中的输入字段）。
max_alternatives	integer	

decisionlistnode 属性	值	属性说明
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

discriminantnode 属性



判别式分析所做的假设比 logistic 回归的假设更严格，但在符合这些假设时，判别式分析可以作为 logistic 回归分析的有用替代项或补充。有关详细信息，请参阅第 10 章中的判别式节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create discriminantnode
set :discriminantnode.target = custcat
set :discriminantnode.use_partitioned_data = False
set :discriminantnode.method = Stepwise
```

discriminantnode 属性	值	属性说明
target	field	Discriminant 模型使用单个目标字段，以及一个或多个输入字段。也不使用加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	flag	“高级输出”对话框中的统计选项。
univariate_anovas	flag	
box_m	flag	
within_group_covariance	flag	
within_groups_correlation	flag	
separate_groups_covariance	flag	
total_covariance	flag	
fishers	flag	
unstandardized	flag	
casewise_results	flag	“高级输出”对话框中的分类选项。
limit_to_first	number	默认值为 10。
summary_table	flag	

discriminantnode 属性	值	属性说明
leave_one_classification	flag	
combined_groups	flag	
separate_groups_covariance	flag	矩阵选项类协方差。
territorial_map	flag	
combined_groups	flag	散点图选项联合组。
separate_groups	flag	散点图选项独立组。
summary_of_steps	flag	
F_pairwise	flag	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	number	
criteria	UseValue UseProbability	
F_value_entry	number	默认值为 3.84。
F_value_removal	number	默认值为 2.71。
probability_entry	number	默认值为 .05。
probability_removal	number	默认值为 .10。
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

factornode 属性



因子/主成分分析节点提供了用于降低数据复杂程度的强大数据缩减技术。主成份分析 (PCA) 可找出输入字段的线性组合, 该组合最好地捕获了整个字段集合中的方差, 且组合中的各个成分相互正交 (相互垂直)。因子分析则尝试识别底层因素, 这些因素说明了观测的字段集合内的相关性模式。这两种方式的目标都是找到有效概括原始字段集中的信息的一小部分导出字段。有关详细信息, 请参阅第 10 章中的主成份分析/因子节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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

```

factornode 属性	值	属性说明
inputs	[field1 ... fieldN]	PCA/因子模型使用一张输入字段列表, 但没有目标字段。也不使用加权和频率字段。有关详细信息, 请参阅第 171 页码通用建模节点属性。
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	number	
complete_records	flag	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	number	
max_factor	number	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	number	如果选择 DirectOblimin 作为旋转数据的类型, 则可以指定 delta 的值。如果未指定一个值, 则将使用 delta 的默认值。

factornode 属性	值	属性说明
kappa	number	如果选择 Promax 作为旋转数据的类型，则可以指定 kappa 的值。如果未指定一个值，则将使用 kappa 的默认值。
sort_values	flag	
hide_values	flag	
hide_below	number	

featureselectionnode 属性



“特征选择”节点会根据某组条件（例如缺失值百分比）筛选可删除的输入字段；对于保留的输入，将相对于指定目标对其重要性进行排序。例如，假如某个给定数据集有上千个潜在输入，那么哪些输入最有可能用于对患者结果进行建模呢？有关详细信息，请参阅第 4 章中的特征选择节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

关于创建和应用“特征选择”模型的详细示例，请参阅第 12 页码 页第 2 章 章中的 独立脚本示例：生成特征选择模型。

featureselectionnode 属性	值	属性说明
target	field	特征选择模型根据指定目标选择预测变量。也不使用加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
screen_single_category	flag	如果为 True，则将筛选相对于记录总数而言同个类别中具有过多记录的字段。
max_single_category	number	指定 screen_single_category 为 True 时使用的阈值。
screen_missing_values	flag	如果为 True，则将筛选具有过多缺失值的字段，字段数表示为记录总数的百分比。
max_missing_values	number	

featureselectionnode 属性	值	属性说明
screen_num_categories	flag	如果为 True, 则将筛选相对于记录总数而言具有过多类别的字段。
max_num_categories	number	
screen_std_dev	flag	如果为 True, 则将筛选标准差小于或等于指定最小值的字段。
min_std_dev	number	
screen_coeff_of_var	flag	如果为 True, 则将筛选方差系数小于或等于指定最小值的字段。
min_coeff_of_var	number	
criteria	Pearson Likelihood CramersV Lambda	对照分类目标对分类预测变量进行排序时, 指定用作确定重要性值的根据的度量。
unimportant_below	number	指定用于将变量分级为“重要”、“一般”或“不重要”的 p 阈值。接受从 .0 到 1.0 之间的值。
important_above	number	接受从 .0 到 1.0 之间的值。
unimportant_label	string	指定“不重要”级别的标签。
marginal_label	string	
important_label	string	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“重要”字段。
select_marginal	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“边际”字段。
select_unimportant	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“不重要”字段。
importance_value	number	在 selection_mode 设置为 ImportanceValue 时, 指定要使用的截断值。接受从 0 到 100 之间的值。
top_n	integer	在 selection_mode 设置为 TopN 时, 指定要使用的截断值。接受从 0 到 1000 之间的值。

genlinnode 属性



“广义线性”模型对一般线性模型进行了扩展, 这样因变量通过指定的关联函数与因子和协变量线性相关。另外, 该模型允许因变量呈非正态分布。它包括统计模型大部分的功能, 其中包括线性回归、logistic 回归、用于计数数据的对数线性模型以及区间删失生存模型。有关详细信息, 请参阅第 10 章中的 GenLin 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create genlinnode
set :genlinnode.model_type = MainAndAllTwoWayEffects
set :genlinnode.offset_type = Variable
set :genlinnode.offset_field = Claimant
```

genlinnode 属性	值	属性说明
target	field	广义线性模型要求单个目标字段（必须是一个名义或标志字段），以及一个或多个输入字段。还可以指定加权字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
use_weight	flag	
weight_field	field	字段类型仅为连续。
target_represents_trials	flag	
trials_type	Variable FixedValue	
trials_field	field	字段类型为连续、标志或有序。
trials_number	number	默认值为 10。
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	field	字段类型仅为连续。
offset_value	number	必须为实数。
base_category	Last First	
include_intercept	flag	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: 逆高斯。 NEGBIN: 负二项式。
negbin_para_type	Specify Estimate	
negbin_parameter	number	默认值为 1。必须含有非负实数。
tweedie_parameter	number	

genlincode 属性	值	属性说明
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: 互补双对数。 LOGC: 对数补数。 NEGBIN: 负二项式。 NLOGLOG: 负双对数。 CUMCAUCHIT: 累积 Cauchit。 CUMCLOGLOG: 累积互补双对数。 CUMLOGIT: 累积 logit。 CUMNLOGLOG: 累积负双对数。 CUMPROBIT: 累积 probit。
power	number	值必须为非 0 的实数。
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	number	默认值为 1; 仅允许使用正整数。
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	number	默认值为 1; 必须大于 0。
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	number	默认值为 100; 仅允许使用非负整数。
max_step_halving	number	默认值为 5; 仅允许使用正整数。
check_separation	flag	
start_iteration	number	默认值为 20; 仅允许使用正整数。
estimates_change	flag	
estimates_change_min	number	默认值为 1E-006; 仅允许使用正数。
estimates_change_type	Absolute Relative	
loglikelihood_change	flag	
loglikelihood_change_min	number	仅允许使用整数。
loglikelihood_change_type	Absolute Relative	
hessian_convergence	flag	
hessian_convergence_min	number	仅允许使用整数。
hessian_convergence_type	Absolute Relative	
case_summary	flag	
contrast_matrices	flag	
descriptive_statistics	flag	
estimable_functions	flag	

genlnode 属性	值	属性说明
model_info	flag	
iteration_history	flag	
goodness_of_fit	flag	
print_interval	number	默认值为 1；必须为正整数。
model_summary	flag	
lagrange_multiplier	flag	
parameter_estimates	flag	
include_exponential	flag	
covariance_estimates	flag	
correlation_estimates	flag	
analysis_type	TypeI TypeIII TypeIAndTypeIII	
statistics	Wald LR	
citype	Wald Profile	
tolerancelevel	number	默认值为 .0001。
confidence_interval	number	默认值为 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	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

GLMM 节点属性



广义线性混合模型 (GLMM) 扩展了线性模型，使得目标可以有非正态分布，通过指定的连接函数与因子和协变量线性相关，并且观测值可能相关。广义线性混合模型涵盖了从简单线性回归到复杂的非正态纵向数据多变量模型的各种模型。有关详细信息，请参阅第 10 章中的 GLMM 节点中的 IBM SPSS Modeler 15 建模节点。

glimmnode 属性	值	属性说明
residual_subject_spec	structured	唯一定义数据集内主体的指定分类字段值的组合。
repeated_measures	structured	用于标识重复观测的字段。
residual_group_spec	[field1 ... fieldN]	定义独立的重复效应协方差参数集合的字段。
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	为残差指定协方差结构。
custom_target	flag	指明是使用在上游节点定义的目标 (<code>false</code>) 还是由 <code>target_field</code> 指定的自定义目标 (<code>true</code>)。
target_field	field	<code>custom_target</code> 为 <code>true</code> 时用作目标的字段。
use_trials	flag	指明当目标响应是发生在一组试验中的事件数时, 要使用附加字段还是指定试验数的值。默认值是 <code>false</code> 。
use_field_or_value	Field Value	指明要使用字段 (默认) 还是值来指定试验数。
trials_field	field	用于指定试验数的字段。
trials_value	integer	用于指定试验数的值。如已指定, 最小值为 1。
use_custom_target_reference	flag	指明是否对分类目标使用自定义参考类别。默认值是 <code>false</code> 。
target_reference_value	string	<code>use_custom_target_reference</code> 为 <code>true</code> 时使用的参考类别。
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	目标值分布的通用模型。从 <code>target_distribution</code> 提供的列表中选择 <code>Custom</code> 以指定分布。
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	当 <code>dist_link_combination</code> 为 <code>Custom</code> 时目标值的分布。

glimmnode 属性	值	属性说明
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	将目标值关联到预测变量的联接函数。
link_function_param	number	要使用的联接函数参数值。仅于 link_function 为 POWER 时适用。
use_predefined_inputs	flag	指明固定效应字段是定义为输入字段的上流 (true) 还是来自 fixed_effects_list (false)。默认值是 false。
fixed_effects_list	structured	如果 use_predefined_inputs 为 false, 则指定将输入字段用作固定效应字段。
use_intercept	flag	如果 true (默认), 则在模型中包含截距。
random_effects_list	structured	指定为随机效应的字段列表。
regression_weight_field	field	用作分析权重字段的字段。
use_offset	None offset_value offset_field	表明如何指定平移。值 None 表示没有使用平移。
offset_value	number	use_offset 设置为 offset_value 时用于平移的值。
offset_field	field	use_offset 设置为 offset_field 时用于平移值的字段。
target_category_order	Ascending Descending Data	分类目标的排序顺序。值 Data 指定使用数据中找到的排序顺序。默认值是 Ascending。
inputs_category_order	Ascending Descending Data	分类预测变量的排序顺序。值 Data 指定使用数据中找到的排序顺序。默认值是 Ascending。
max_iterations	integer	算法将执行的最大迭代次数。非负整数; 默认值为 100。
confidence_level	integer	用于计算模型系数的区间估计值的置信水平。非负整数; 最大值为 100, 默认值为 95。
degrees_of_freedom_method	Fixed Varied	用来指定如何为显著性检验计算自由度。
test_fixed_effects_coefficients	Model Robust	计算参数估计值协方差矩阵的方法。
use_model_name	flag	表明是为模型指定自定义名称 (true) 还是使用系统生成的名称 (false)。默认值是 false。
model_name	string	如果 use_model_name 为 true, 则指定使用的模型名称。

glimmnode 属性	值	属性说明
confidence	highest difference	计算评分置信度值的基准：最高预测概率或最高与第二高预测概率之差。
score_category_probabilities	flag	如果为 true，则为分类目标生成预测概率。默认值是 false。
max_categories	integer	如果 score_category_probabilities 为 true，则指定保存最大类别数。
score_propensity	flag	如果为 true，则为标记目标字段生成倾向得分，指示字段结果为“true”的可能性。
emeans	structure	对于固定效应列表中的每个分类字段，指定是否生成估计边际均值。
covariance_list	structure	对于固定效应列表中的每个连续字段，指定计算估计边际均值时是使用均值还是自定义值。
mean_scale	Original Transformed	用于指定是基于目标原始尺度（默认）还是基于联接函数转换计算估计边际均值。
comparison_adjustment_method	LSD SEQBONFERRONI SEQSIDAK	使用多个对比执行假设检验时使用的调整方法。

kmeansnode 属性



K-Means 节点将数据集聚类到不同分组（或聚类）。此方法将定义固定的聚类数量，将记录迭代分配给聚类，以及调整聚类中心，直到进一步优化无法再改进模型。k-means 节点作为一种非监督学习机制，它并不试图预测结果，而是揭示隐含在输入字段集中的模式。有关详细信息，请参阅第 11 章中的 K-Means 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

kmeansnode 属性	值	属性说明
inputs	[field1 ... fieldN]	K-means 模型在一系列输入字段上执行聚类分析，但并不使用目标字段。也不使用加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
num_clusters	number	
gen_distance	flag	
cluster_label	String Number	
label_prefix	string	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	number	
tolerance	number	
encoding_value	number	
optimize	Speed Memory	用于指定优化建模的方式是速度还是内存。

knnnode 属性



The k-最近相邻元素 (KNN) 节点将新的个案关联到预测变量空间中与其最邻近的 k 个对象的类别或值（其中 k 为整数）。类似个案相互靠近，而不同个案相互远离。有关详细信息，请参阅第 16 章中的 KNN 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

knnnode 属性	值	属性说明
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	flag	

knnnode 属性	值	属性说明
use_case_labels	flag	选中复选框启用下一个选项。
case_labels_field	field	
identify_focal_cases	flag	选中复选框启用下一个选项。
focal_cases_field	field	
automatic_k_selection	flag	
fixed_k	integer	仅当 automatic_k_selection 为 False 时启用。
minimum_k	integer	仅当 automatic_k_selection 为 True 时启用。
maximum_k	integer	
distance_computation	Euclidean CityBlock	
weight_by_importance	flag	
range_predictions	Mean Median	
perform_feature_selection	flag	
forced_entry_inputs	[field1 ... fieldN]	
stop_on_error_ratio	flag	
number_to_select	integer	
minimum_change	number	
validation_fold_assignment_by_field	flag	
number_of_folds	integer	仅当 validation_fold_assignment_by_field 为 False 时启用。
set_random_seed	flag	
random_seed	number	
folds_field	field	仅当 validation_fold_assignment_by_field 为 True 时启用。
all_probabilities	flag	
save_distances	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

kohonenode 属性



Kohonen 节点会生成一种神经网络，此神经网络可用于将数据集聚类到各个差异组。此网络训练完成后，相似的记录应在输出映射中紧密地聚集，差异大的记录则应彼此远离。您可以通过查看模型块中每个单元所捕获观测值的数量来找出规模较大的单元。这将让您对聚类的相应数量有所估计。有关详细信息，请参阅第 11 章中的 Kohonen 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```

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

```

kohonennode 属性	值	属性说明
inputs	[field1 ... fieldN]	Kohonen 模型使用一张输入字段列表，但没有目标字段。不使用频率和加权字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
continue	flag	
show_feedback	flag	
stop_on	Default Time	
time	number	
optimize	Speed Memory	用于指定优化建模的方式是速度还是内存。
cluster_label	flag	
mode	Simple Expert	
width	number	
length	number	
decay_style	Linear Exponential	
phase1_neighborhood	number	
phase1_eta	number	
phase1_cycles	number	
phase2_neighborhood	number	
phase2_eta	number	
phase2_cycles	number	

linearnode 属性



线性回归模型根据目标与一个或多个预测变量间的线性关系来预测连续目标。有关详细信息，请参阅第 10 章中的线性模型中的 IBM SPSS Modeler 15 建模节点。

示例

```
create linearnode
# Build Options tab - Objectives panel
set:linearnode.objective = Standard
# Build Options tab - Model Selection panel
set:linearnode.model_selection = BestSubsets
set:linearnode.criteria_best_subsets = ASE
# Build Options tab - Ensembles panel
set:linearnode.combining_rule_categorical = HighestMeanProbability
```

linearnode 属性	值	属性说明
target	字段	指定一个目标字段。
inputs	[字段1 ... 字段N]	模型使用的预测变量字段。
continue_training_existing_model	标志	
objective	Standard Bagging Boosting psm	psm 用于非常大的数据集，同时需要 Server 连接。
use_auto_data_preparation	标志	
confidence_level	数字	
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	数字	
probability_removal	数字	
use_max_effects	标志	
max_effects	数字	
use_max_steps	标志	
max_steps	数字	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuous	Mean Median	
component_models_n	数字	
use_random_seed	标志	

linearnode 属性	值	属性说明
random_seed	数字	
use_custom_model_name	标志	
custom_model_name	字符串	
use_custom_name	标志	
custom_name	字符串	
tooltip	字符串	
keywords	字符串	
annotation	字符串	

logregnode 属性



Logistic 回归是一种统计方法，它可根据输入字段的值对记录进行分类。它类似于线性回归，但采用的是类别目标字段而非数字范围。[有关详细信息，请参阅第 10 章中的逻辑节点中的 IBM SPSS Modeler 15 建模节点。](#)

多项式示例

```
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
```

二项式示例

```
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

```

logregnode 属性	值	属性说明
target	field	Logistic 回归模型使用单个目标字段，以及一个或多个输入字段。不使用频率和加权字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
logistic_procedure	Binomial Multinomial	
include_constant	flag	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	将 FullFactorial 指定为模型类型时，即使指定了步进方法，步进方法也不会运行。而是使用 Enter 方法。 如果将模型类型设置为 Custom，但未指定自定义字段，则将构建主效应模型。
custom_terms	[{BP 性别} {BP} {年龄}]	示例： set :logregnode. custom_terms = [{Na} {K} {Na K}]
multinomial_base_category	string	指定如何确定参考类别。
binomial_categorical_input	string	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	用于指定确定对比方式的分类输入的键控属性。 用法格式： NODE.binomial_input_contrast.FIELDNAME
binomial_input_category	First Last	用于指定确定引用类别方式的分类输入的键控属性。 用法格式： NODE.binomial_input_category.FIELDNAME
scale	None UserDefined Pearson Deviance	
scale_value	number	

logregnode 属性	值	属性说明
all_probabilities	flag	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	number	
use_max_terms	flag	
max_terms	number	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	number	
probability_removal	number	
binomial_probability_entry	number	
binomial_probability_removal	number	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	number	
max_steps	number	
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	number	
iteration_history	flag	
history_steps	number	
summary	flag	
likelihood_ratio	flag	
asymptotic_correlation	flag	
goodness_fit	flag	
parameters	flag	
confidence_interval	number	
asymptotic_covariance	flag	
classification_table	flag	
stepwise_summary	flag	

logregnode 属性	值	属性说明
info_criteria	flag	
monotonicity_measures	flag	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	flag	
binomial_parameters	flag	
binomial_iteration_history	flag	
binomial_classification_plots	flag	
binomial_ci_enable	flag	
binomial_ci	number	
binomial_residual	outliers all	
binomial_residual_enable	flag	
binomial_outlier_threshold	number	
binomial_classification_cutoff	number	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	

neuralnetnode 属性

警告：在此发行版中提供了具有增强功能的新版本的神经网络建模节点，并将在下一节（neuralnetwork）中进行介绍。尽管您仍然可以使用先前版本来构建模型并对其评分，但我们建议您更新脚本以使用新的版本。此处保留之前版本的详细信息以供参考。

示例

```
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
```

neuralnetnode 属性	值	属性说明
targets	[field1 ... fieldN]	Neural Net 节点需要一个或多个目标字段以及一个或多个输入字段。忽略频率和加权字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	flag	
train_pct	number	
set_random_seed	flag	
random_seed	number	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	停止模式。
accuracy	number	停止时的准确性。
cycles	number	要训练的周期。
time	number	要训练的时间（分钟）。
continue	flag	
show_feedback	flag	
binary_encode	flag	
use_last_model	flag	
gen_logfile	flag	
logfile_name	string	
alpha	number	
initial_eta	number	
high_eta	number	
low_eta	number	
eta_decay_cycles	number	
hid_layers	One Two Three	
hl_units_one	number	
hl_units_two	number	
hl_units_three	number	
persistence	number	

neuralnetnode 属性	值	属性说明
m_topologies	string	
m_non_pyramids	flag	
m_persistence	number	
p_hid_layers	One Two Three	
p_hl_units_one	number	
p_hl_units_two	number	
p_hl_units_three	number	
p_persistence	number	
p_hid_rate	number	
p_hid_pers	number	
p_inp_rate	number	
p_inp_pers	number	
p_overall_pers	number	
r_persistence	number	
r_num_clusters	number	
r_eta_auto	flag	
r_alpha	number	
r_eta	number	
optimize	Speed Memory	用于指定优化建模的方式是速度还是内存。
calculate_variable_importance	flag	注意：以前版本中使用的 <code>sensitivity_analysis</code> 属性已由此属性代替。仍然支持旧属性，但建议使用 <code>calculate_variable_importance</code> 。
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

neuralnetworknode 属性



神经网络节点使用的模型是对人类大脑处理信息的方式简化了的模型。此模型通过模拟大量类似于神经元的抽象形式的互连简单处理单元而运行。神经网络是功能强大的一般函数估计器，只需要最少的统计或数学知识就可以对其进行训练或应用。

示例

```
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

```

neuralnetworknode 属性	值	属性说明
targets	[字段1 ... 字段N]	指定目标字段。
inputs	[字段1 ... 字段N]	模型使用的预测变量字段。
splits	[字段1 ... 字段N]	指定拆分建模使用的字段。
use_partition	标志	如果定义了分区字段，则此选项可确保仅训练分区的数据用于构建模型。
continue	标志	继续训练现有模型。
objective	Standard Bagging Boosting psm	psm 用于非常大的数据集，同时需要 Server 连接。
method	Multi layerPerceptron RadialBasisFunction	
use_custom_layers	标志	
first_layer_units	数字	
second_layer_units	数字	
use_max_time	标志	
max_time	数字	
use_max_cycles	标志	
max_cycles	数字	
use_min_accuracy	标志	
min_accuracy	数字	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	数字	
overfit_prevention_pct	数字	
use_random_seed	标志	
random_seed	数字	
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	标志	
custom_model_name	字符串	
confidence	onProbability onIncrease	
score_category_probabilities	标志	
max_categories	数字	
score_propensity	标志	
use_custom_name	标志	

neuralnetworknode 属性	值	属性说明
custom_name	字符串	
tooltip	字符串	
keywords	字符串	
annotation	字符串	

questnode 属性



QUEST 节点可提供用于构建决策树的二元分类法，此方法的设计目的是减少大型 C&R 树分析所需的处理时间，同时也减少在分类树方法中发现的趋势以便支持允许有多个分割的输入。输入字段可以是数字范围（连续），但目标字段必须是分类。所有分割都是二元的。[有关详细信息，请参阅第 6 章中的 QUEST 节点中的 IBM SPSS Modeler 15 建模节点。](#)

示例

```
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
```

questnode 属性	值	属性说明
target	field	QUEST 模型要求使用单个目标字段，以及一个或多个输入字段。还可以指定频率字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
continue_training_existing_model	flag	
objective	Standard Boosting Bagging psm	psm 用于非常大的数据集，同时需要 Server 连接。
model_output_type	Single InteractiveBuilder	
use_tree_directives	flag	

questnode 属性	值	属性说明
tree_directives	string	
use_max_depth	Default Custom	
max_depth	integer	最大树深度从 0 到 1000。只在 use_max_depth = Custom 时使用。
prune_tree	flag	修剪树以防止过拟合。
use_std_err	flag	使用最大风险差值（标准误）。
std_err_multiplier	number	最大差值。
max_surrogates	number	最大代用项。
use_percentage	flag	
min_parent_records_pc	number	
min_child_records_pc	number	
min_parent_records_abs	number	
min_child_records_abs	number	
use_costs	flag	
costs	structured	使用以下格式的结构化属性： [{drugA drugB 1.5} {drugA drugC 2.1}]，其中花括号 ({}) 中的参数为实际预测成本。
priors	Data Equal Custom	
custom_priors	structured	使用以下格式的结构化属性： set :cartnode. custom_priors = [{ drugA 0.3 } { drugB 0.6 }]
adjust_priors	flag	
trails	number	用于推进或 bagging 的成分模型数。
set_ensemble_method	Voting HighestProbability HighestMeanProbability	分类目标的默认组合规则。
range_ensemble_method	Mean Median	连续目标的默认组合规则。
large_boost	flag	将推进应用到大型数据集。
split_alpha	number	用于分割的显著性水平。
train_pct	number	防止过度拟合集合。
set_random_seed	flag	重复结果选项。
seed	number	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

regressionnode 属性



线性回归是一种通过拟合直线或平面以实现汇总数据和预测的普通统计方法，它可使预测值和实际输出值之间的差异最小化。

注意：在将来的版本中，回归节点将被线性节点替换。我们建议从现在开始为线性回归使用线性模型。

示例

```
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
```

regressionnode 属性	值	属性说明
target	field	回归模型要求单个目标字段，以及一个或多个输入字段。还可以指定加权字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
method	Enter Stepwise Backwards Forwards	
include_constant	flag	
use_weight	flag	
weight_field	field	
mode	Simple Expert	
complete_records	flag	
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	对实参使用双引号。
stepping_method	useP useF	useP：使用 F 的概率 useF：使用 F 值
probability_entry	number	
probability_removal	number	
F_value_entry	number	
F_value_removal	number	
selection_criteria	flag	
confidence_interval	flag	
covariance_matrix	flag	
collinearity_diagnostics	flag	
regression_coefficients	flag	
exclude_fields	flag	
durbin_watson	flag	
model_fit	flag	
r_squared_change	flag	
p_correlations	flag	
descriptives	flag	
calculate_variable_importance	flag	

sequencenode 属性



序列节点可发现连续数据或与时间有关的数据中的关联规则。序列是一系列可能会以可预测顺序发生的项目集合。例如，一个购买了剃刀和须后水的顾客可能在下次购物时购买剃须膏。序列节点基于 CARMA 关联规则算法，该算法使用有效的两步法来发现序列。有关详细信息，请参阅第 12 章中的序列节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
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
```

sequencenode 属性	值	属性说明
id_field	field	要创建序列模型，您需要指定一个 ID 字段以及一个可选的时间字段，以及一个或多个内容字段。也不使用加权和频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
time_field	field	
use_time_field	flag	
content_fields	[字段1 ... 字段n]	
contiguous	flag	
min_supp	number	

sequencenode 属性	值	属性说明
min_conf	number	
max_size	number	
max_predictions	number	
mode	Simple Expert	
use_max_duration	flag	
max_duration	number	
use_gaps	flag	
min_item_gap	number	
max_item_gap	number	
use_pruning	flag	
pruning_value	number	
set_mem_sequences	flag	
mem_sequences	integer	

slrmnode 属性



自学响应模型 (SLRM) 节点可用于构建一个包含单个新观测值或少量新观测值的模型，通过此模型，无需使用全部数据对模型进行重新训练即可对模型进行重新评估。有关详细信息，请参阅第 14 章中的 SLRM 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create slrmnode
set :slrmnode.target = Offer
set :slrmnode.target_response = Response
set :slrmnode.inputs = ['Cust_ID' 'Age' 'Ave_Bal']
```

slrmnode 属性	值	属性说明
target	field	目标字段必须是名义或标志字段。还可以指定频率字段。有关详细信息，请参阅第 171 页码通用建模节点属性。
target_response	field	类型必须是标志。
continue_training_existing_model	flag	
target_field_values	flag	使用全部：使用来自源的全部值。 指定：选择所需的值。
target_field_values_specify	[field1 ... fieldN]	
include_model_assessment	flag	
model_assessment_random_seed	number	必须为实数。
model_assessment_sample_size	number	必须为实数。

slrmnode 属性	值	属性说明
model_assessment_iterations	number	迭代数。
display_model_evaluation	flag	
max_predictions	number	
randomization	number	
scoring_random_seed	number	
sort	Ascending Descending	指定先显示得分最高还是最低的报价。
model_reliability	flag	
calculate_variable_importance	flag	

statisticsmodelnode 属性



Statistics 模型节点使您能够通过运行生成 PMML 的 IBM® SPSS® Statistics 过程分析和处理数据。此节点需要 SPSS Statistics 的许可副本。有关详细信息，请参阅第 8 章中的 Statistics 模型节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

有关此节点属性的信息，请参阅 [statisticsmodelnode 属性第 279 页码](#)。

svmnode 属性



使用 Support Vector Machine (SVM) 节点，可以将数据分为两组，而无需过度拟合。SVM 可以与大量数据集配合使用，如那些含有大量输入字段的数据集。有关详细信息，请参阅第 15 章中的 SVM 节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create svmnode
# Expert tab
set :svmnode.mode=Expert
set :svmnode.all_probabilities=True
set :svmnode.kernel=Polynomial
set :svmnode.gamma=1.5
```

svmnode 属性	值	属性说明
all_probabilities	flag	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (默认) 1.0E-4 1.0E-5 1.0E-6	确定何时停止优化算法。
regularization	number	又称为 C 参数。

svmnode 属性	值	属性说明
precision	number	仅当目标字段的测量级别为 Continuous 时才使用。
kernel	RBF (默认) Polynomial Sigmoid Linear	用于变换的核函数的类型。
rbf_gamma	number	仅当 kernel 为 RBF 时使用。
gamma	number	仅当 kernel 为 Polynomial 或 Sigmoid 时使用。
bias	number	
degree	number	仅当 kernel 为 Polynomial 时使用。
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

timeseriesnode 属性



时间序列节点估计时间序列数据的指数平滑模型、单变量自回归整合移动平均 (ARIMA) 模型和多变量 ARIMA (即变换函数) 模型, 并生成未来性能的预测数据。在时间序列节点之前必须有时间区间节点。有关详细信息, 请参阅第 13 章中的时间序列建模节点中的 IBM SPSS Modeler 15 建模节点。

示例

```
create timeseriesnode
set :timeseriesnode.method = Exsmooth
set :timeseriesnode.exsmooth_model_type = HoltsLinearTrend
set :timeseriesnode.exsmooth_transformation_type = None
```

timeseriesnode 属性	值	属性说明
targets	field	时间序列节点预测了一个或多个目标, 可选择使用一个或多个输入字段作为预测变量。不使用频率和加权字段。有关详细信息, 请参阅第 171 页码通用建模节点属性。
continue	flag	
method	ExpertModeler Exsmooth Arima Reuse	

timeseriesnode 属性	值	属性说明
expert_modeler_method	flag	
consider_seasonal	flag	
detect_outliers	flag	
expert_outlier_additive	flag	
expert_outlier_level_shift	flag	
expert_outlier_innovational	flag	
expert_outlier_level_shift	flag	
expert_outlier_transient	flag	
expert_outlier_seasonal_additive	flag	
expert_outlier_local_trend	flag	
expert_outlier_additive_patch	flag	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma_p	integer	
arma_d	integer	
arma_q	integer	
arma_sp	integer	
arma_sd	integer	
arma_sq	integer	
arma_transformation_type	None SquareRoot NaturalLog	
arma_include_constant	flag	
tf_arma_p.fieldname	integer	用于变换函数。
tf_arma_d.fieldname	integer	用于变换函数。
tf_arma_q.fieldname	integer	用于变换函数。
tf_arma_sp.fieldname	integer	用于变换函数。
tf_arma_sd.fieldname	integer	用于变换函数。
tf_arma_sq.fieldname	integer	用于变换函数。
tf_arma_delay.fieldname	integer	用于变换函数。
tf_arma_transformation_type.fieldname	None SquareRoot NaturalLog	用于变换函数。
arma_detect_outlier_mode	None Automatic	
arma_outlier_additive	flag	
arma_outlier_level_shift	flag	
arma_outlier_innovational	flag	

timeseriesnode 属性	值	属性说明
arima_outlier_transient	flag	
arima_outlier_seasonal_additive	flag	
arima_outlier_local_trend	flag	
arima_outlier_additive_patch	flag	
conf_limit_pct	real	
max_lags	integer	
events	字段	
scoring_model_only	flag	用于具有大量（数万个）时间序列的模型

twostepnode 属性



TwoStep 节点使用两步聚类方法。第一步完成简单数据处理，以便将原始输入数据压缩为可管理的子聚类集合。第二步使用层级聚类方法将子聚类一步一步合并为更大的聚类。TwoStep 具有一个优点，就是能够为训练数据自动估计最佳聚类数。它可以高效处理混合的字段类型和大型的数据集。[有关详细信息，请参阅第 11 章中的两步聚类节点中的 IBM SPSS Modeler 15 建模节点。](#)

示例

```
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
```

twostepnode 属性	值	属性说明
inputs	[field1 ... fieldN]	两步模型使用一张输入字段列表，但没有目标字段。也不识别加权和频率字段。 有关详细信息，请参阅第 171 页码通用建模节点属性。
standardize	flag	
exclude_outliers	flag	
percentage	number	
cluster_num_auto	flag	
min_num_clusters	number	
max_num_clusters	number	
num_clusters	number	

twostepnode 属性	值	属性说明
cluster_label	String Number	
label_prefix	string	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

模型块节点属性

模型块节点具有与其他节点相同的通用属性。有关详细信息，请参阅第 100 页码第 9 章中的通用节点属性。

applyanomalydetectionnode 属性

可使用异常检测建模节点生成异常检测模型块。该模型块的脚本名称为 applyanomalydetectionnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 171 页码中的anomalydetectionnode 属性。

applyanomalydetectionnode 属性	值	属性说明
anomaly_score_method	FlagAndScore FlagOnly ScoreOnly	确定创建哪些输出用于评分。
num_fields	整数	要报告的字段。
discard_records	标志	指示是否从输出中丢弃记录。
discard_anomalous_records	标志	指示是丢弃异常记录还是丢弃非异常记录。默认状态为 off，表示丢弃非异常记录。否则，如果状态为 on，则丢弃异常记录。仅当启用 discard_records 属性时，才会启用此属性。

applyapriorinode 属性

可使用 Apriori 建模节点生成 Apriori 模型块。该模型块的脚本名称为 applyapriorinode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 172 页码中的apriorinode 属性。

applyapriorinode 属性	值	属性说明
max_predictions	数值（整数）	
ignore_unmatched	标志	
allow_repeats	标志	
check_basket	NoPredictions Predictions NoCheck	
criterion	置信度 支持度 RuleSupport Lift 部署能力	

applyautoclassifiernode 属性

“自动分类器”建模节点可用于生成“自动分类器”模型块。该模型块的脚本名称为 applyautoclassifiernode。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 174 页码中的 autoclassifiernode 属性。](#)

applyautoclassifiernode 属性	值	属性说明
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	指定用于确定整体得分的方法。仅当选定的目标为标志字段时，才会应用该设置。
flag_voting_tie_selection	Random HighestConfidence RawPropensity	如果已选定投票方法，则指定解决结的方法。仅当选定的目标为标志字段时，才会应用该设置。
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	指定用于确定整体得分的方法。仅当选定的目标为集合字段时，才会应用该设置。
set_voting_tie_selection	Random HighestConfidence	如果已选定投票方法，则指定解决结的方法。仅当选定的目标为名义字段时，才会应用该设置。

applyautoclusternode 属性

“自动聚类”建模节点可用于生成“自动聚类”模型块。该模型块的脚本名称为 applyautoclusternode。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 176 页码中的 autoclusternode 属性。](#)

applyautonumericnode 属性

“自动数值”建模节点可用于生成“自动数值”模型块。该模型块的脚本名称为 applyautonumericnode。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 178 页码中的 autonumericnode 属性。](#)

applyautonumericnode 属性	值	属性说明
calculate_standard_error	标志	

applybayesnetnode 属性

可使用贝叶斯网络建模节点生成贝叶斯网络模型块。该模型块的脚本名称为 `applybayesnetnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 179 页码中的 `bayesnetnode` 属性。

applybayesnetnode 属性	值	属性说明
<code>all_probabilities</code>	标志	
<code>raw_propensity</code>	标志	
<code>adjusted_propensity</code>	标志	
<code>calculate_raw_propensities</code>	标志	
<code>calculate_adjusted_propensities</code>	标志	

applyc50node 属性

可使用 C5.0 建模节点生成 C5.0 模型块。该模型块的脚本名称为 `applyc50node`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 180 页码中的 `c50node` 属性。

applyc50node 属性	值	属性说明
<code>sql_generate</code>	Never NoMissingValues	用于设置规则集执行期间的 SQL 生成选项。
<code>calculate_conf</code>	标志	启用 SQL 生成时可用；此属性将置信度的计算包括在生成的树中。
<code>calculate_raw_propensities</code>	标志	
<code>calculate_adjusted_propensities</code>	标志	

applycarmanode 属性

可使用 CARMA 建模节点生成 CARMA 模型块。该模型块的脚本名称为 `applycarmanode`。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 181 页码中的 `carmanode` 属性。

applycartnode 属性

可使用 C&R 树建模节点生成 C&R 树模型块。该模型块的脚本名称为 `applycartnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 183 页码中的 `cartnode` 属性。

applycartnode 属性	值	属性说明
<code>sql_generate</code>	Never MissingValues NoMissingValues	用于设置规则集执行期间的 SQL 生成选项。
<code>calculate_conf</code>	标志	启用 SQL 生成时可用；此属性将置信度的计算包括在生成的树中。

applycartnode 属性	值	属性说明
display_rule_id	标志	在得分输出中添加一个字段，表示每个记录分配到的终端节点的 ID。
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applychaidnode 属性

可使用 CHAID 建模节点生成 CHAID 模型块。该模型块的脚本名称为 applychaidnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 185 页码中的 [chaidnode 属性](#)。

applychaidnode 属性	值	属性说明
sql_generate	Never MissingValues	
calculate_conf	标志	
display_rule_id	标志	在得分输出中添加一个字段，表示每个记录分配到的终端节点的 ID。
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applycoxregnode 属性

可使用 Cox 建模节点生成 Cox 模型块。该模型块的脚本名称为 applycoxregnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 187 页码中的 [coxregnode 属性](#)。

applycoxregnode 属性	值	属性说明
future_time_as	区间 字段	
time_interval	数字	
num_future_times	整数	
time_field	字段	
past_survival_time	字段	
all_probabilities	标志	
cumulative_hazard	标志	

applydecisionlistnode 属性

可使用决策列表建模节点生成决策列表模型块。该模型块的脚本名称为 applydecisionlistnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 189 页码中的 decisionlistnode 属性。

applydecisionlistnode 属性	值	属性说明
enable_sql_generation	标志	值为真时，IBM® SPSS® Modeler 会尝试将决策列表模型回推到 SQL。
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applydiscriminantnode 属性

可使用判别式建模节点生成判别式模型块。该模型块的脚本名称为 applydiscriminantnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 190 页码中的 discriminantnode 属性。

applydiscriminantnode 属性	值	属性说明
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applyfactornode 属性

可使用主成分分析/因子建模节点生成主成分分析/因子模型块。该模型块的脚本名称为 applyfactornode。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 191 页码中的 factornode 属性。

applyfeatureselectionnode 属性

可使用特征选择建模节点生成特征选择模型块。该模型块的脚本名称为 applyfeatureselectionnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 193 页码中的 featureselectionnode 属性。

applyfeatureselectionnode 属性	值	属性说明
selected_ranked_fields		指定在模型浏览器中检查哪些已排序字段。
selected_screened_fields		指定在模型浏览器中检查哪些已筛选字段。

applygeneralizedlinearnode 属性

可使用广义线性 (genlin) 建模节点生成广义线性模型块。该模型块的脚本名称为 applygeneralizedlinearnode。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 194 页码中的 genlinnode 属性](#)。

applygeneralizedlinearnode 属性	值	属性说明
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applykmeansnode 属性

可使用 K-Means 建模节点生成 K-Means 模型块。该模型块的脚本名称为 applykmeansnode。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 200 页码中的 kmeansnode 属性](#)。

applyknnnode 属性

KNN 建模节点可用于生成 KNN 模型块。此模型块的脚本名称是 applyknnnode。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 201 页码中的 knnnode 属性](#)。

applyknnnode 属性	值	属性说明
all_probabilities	标志	
save_distances	标志	

applykohonennode 属性

可使用 Kohonen 建模节点生成 Kohonen 模型块。该模型块的脚本名称为 applykohonennode。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 180 页码中的 c50node 属性](#)。

applylinearnode 属性

可使用线性建模节点生成线性模型块。该模型块的脚本名称为 applylinearnode。有关编写建模节点自身脚本的详细信息，[请参阅第 16 章第 204 页码中的 linearnode 属性](#)。

linear 属性	值	属性说明
use_custom_name	标志	
custom_name	字符串	
enable_sql_generation	标志	

applylogregnode 属性

可使用 Logistic 回归建模节点生成 Logistic 回归模型块。该模型块的脚本名称为 applylogregnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 205 页码中的 logregnode 属性。

applylogregnode 属性	值	属性说明
calculate_raw_propensities	标志	

applyneuralnetnode 属性

可使用神经网络建模节点生成神经网络模型块。该模型块的脚本名称为 applyneuralnetnode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 209 页码中的 neuralnetnode 属性。

警告：在此发行版中提供了具有增强功能的新版本的神经网络模型块，并将在下一节 (applyneuralnetwork) 中进行介绍。尽管先前版本仍然可用，但我们建议您更新脚本以使用新的版本。此处保留了先前版本的详细信息以供参考，但会在将来的发行版中不再支持。

applyneuralnetnode 属性	值	属性说明
calculate_conf	标志	启用 SQL 生成时可用；此属性将置信度的计算包括在生成的树中。
enable_sql_generation	标志	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	标志	
calculate_adjusted_propensities	标志	

applyneuralnetworknode 属性

可使用神经网络建模节点生成神经网络模型块。该模型块的脚本名称为 applyneuralnetworknode。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 211 页码中的 neuralnetworknode 属性。

applyneuralnetworknode 属性	值	属性说明
use_custom_name	标志	
custom_name	字符串	
confidence	onProbability onIncrease	
score_category_probabilities	标志	
max_categories	数字	
score_propensity	标志	

applyquestnode 属性

可使用 QUEST 建模节点生成 QUEST 模型块。该模型块的脚本名称为 `applyquestnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 213 页码中的 `questnode` 属性。

applyquestnode 属性	值	属性说明
<code>sql_generate</code>	Never MissingValues NoMissingValues	
<code>calculate_conf</code>	标志	
<code>display_rule_id</code>	标志	在得分输出中添加一个字段，表示每个记录分配到的终端节点的 ID。
<code>calculate_raw_propensities</code>	标志	
<code>calculate_adjusted_propensities</code>	标志	

asapplyregressionnode 属性

可使用线性回归建模节点生成线性回归模型块。该模型块的脚本名称为 `applyregressionnode`。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 215 页码中的 `regressionnode` 属性。

applyselflearningnode 属性

可使用自学响应模型 (SLRM) 建模节点生成 SLRM 模型块。该模型块的脚本名称为 `applyselflearningnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 218 页码中的 `slrmnode` 属性。

applyselflearningnode 属性	值	属性说明
<code>max_predictions</code>	数字	
<code>randomization</code>	数字	
<code>scoring_random_seed</code>	数字	
<code>sort</code>	ascending 降序	指定先显示得分最高还是最低的报价。
<code>model_reliability</code>	标志	将“设置”选项卡中的模型可靠性选项考虑在内。

applysequencenode 属性

可使用序列建模节点生成序列模型块。该模型块的脚本名称为 `applysequencenode`。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 217 页码中的 `sequencenode` 属性。

applysvmnode 属性

可使用 SVM 建模节点生成 SVM 模型块。该模型块的脚本名称为 `applysvmnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 219 页码中的 `svmnode` 属性。

applysvmnode 属性	值	属性说明
<code>all_probabilities</code>	标志	
<code>calculate_raw_propensities</code>	标志	
<code>calculate_adjusted_propensities</code>	标志	

applytimeseriesnode 属性

可使用时间序列建模节点生成时间序列模型块。该模型块的脚本名称为 `applytimeseriesnode`。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 220 页码中的 `timeseriesnode` 属性。

applytimeseriesnode 属性	值	属性说明
<code>calculate_conf</code>	标志	
<code>calculate_residuals</code>	标志	

applytwostepnode 属性

可使用两步建模节点生成 TwoStep 模型块。该模型块的脚本名称为 `applytwostepnode`。该模型块不存在其他属性。有关编写建模节点自身脚本的详细信息，请参阅第 16 章第 222 页码中的 `twostepnode` 属性。

数据库建模节点属性

IBM® SPSS® Modeler 支持与多家数据库提供商的数据挖掘和建模工具集成，包括 Microsoft SQL Server Analysis Services、Oracle Data Mining、IBM® DB2® InfoSphere Warehouse 和 IBM® Netezza® Analytics。有关详细信息，请参阅第 2 章中的数据库建模概述中的 IBM SPSS Modeler 15 数据库内数据挖掘指南。可以使用 SPSS Modeler 应用程序自有的数据库算法来构建模型并对模型进行评分。还可以使用本节介绍的属性通过编写脚本来构建和操纵数据库模型。

例如，下面这段脚本说明了如何使用 SPSS Modeler 的脚本编写界面来创建 Microsoft 决策树模型：

```
create mstreenode
rename :mstreenode as msbuilder
set msbuilder.analysis_server_name = 'localhost'
set msbuilder.analysis_database_name = 'TESTDB'
set msbuilder.mode = 'Expert'
set msbuilder.datasourcesource = '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
```

Microsoft 建模的节点属性

Microsoft 建模节点属性

通用属性

以下属性通用于 Microsoft 数据库建模节点。

通用 Microsoft 节点属性	值	属性说明
analysis_database_name	string	Analysis Services 数据库的名称。
analysis_server_name	string	Analysis Services 主机的名称。
use_transactional_data	flag	指定输入数据是表格格式还是交易格式。
inputs	[field field field]	表格数据的输入字段。

通用 Microsoft 节点属性	值	属性说明
target	field	预测字段（对 MS 聚类节点或序列聚类节点不适用）。
unique_field	field	关键字段。
msas_parameters	structured	算法参数。有关详细信息，请参阅第 235 页 算法参数 。
with_drillthrough	flag	具有 Drillthrough 选项。

MS 决策树

没有为 `mstreenode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 聚类

没有为 `msclusternode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 关联规则

以下特定属性可用于类型为 `msassocnode` 的节点：

msassocnode 属性	值	属性说明
id_field	field	标识数据中的每个交易。
trans_inputs	[field field field]	交易数据的输入字段。
transactional_target	field	预测字段（交易数据）。

MS Naive Bayes

没有为 `msbayesnode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 线性回归

没有为 `msregressionnode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 神经网络

没有为 `msneuralnetworknode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS Logistic 回归

没有为 `mslogisticnode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 时间序列

没有为 `mstimeseriesnode` 类型的节点定义具体属性。请参阅本章节开头部分的通用 Microsoft 属性。

MS 序列聚类

以下特定属性可用于类型为 `mssequenceclusternode` 的节点：

mssequenceclusternode 属性	值	属性说明
<code>id_field</code>	field	标识数据中的每个交易。
<code>input_fields</code>	[field field field]	交易数据的输入字段。
<code>sequence_field</code>	field	序列标识符。
<code>target_field</code>	field	预测字段（表格数据）。

算法参数

每种 Microsoft 数据库模型类型均有可使用 `msas_parameters` 属性来设置的特定参数，例如：

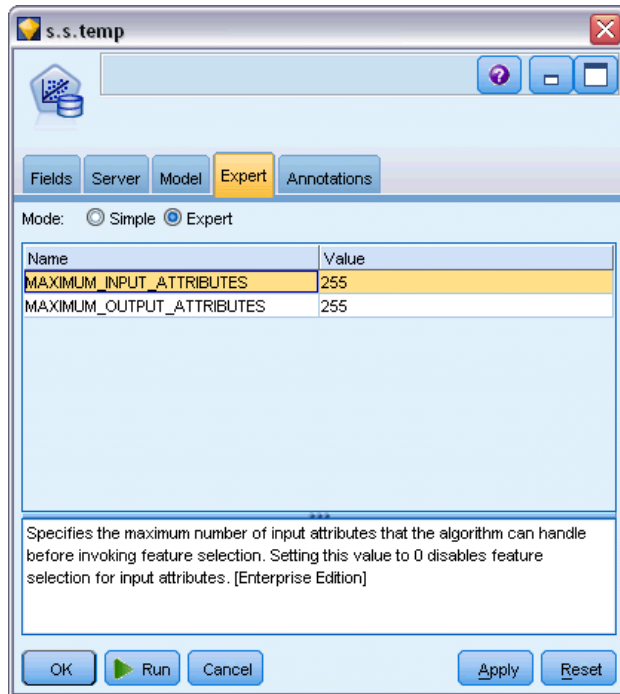
```
set :msregressionnode.msas_parameters =
[["MAXIMUM_INPUT_ATTRIBUTES" 255] ["MAXIMUM_OUTPUT_ATTRIBUTES" 255]]
```

这些参数源自 SQL Server。要查看每个节点的相关参数，请执行如下操作：

- ▶ 将数据库源节点置于工作区。
- ▶ 打开该数据库源节点。
- ▶ 从数据源下拉列表选择一个有效源。
- ▶ 从表名列表选择一个有效表。
- ▶ 单击确定关闭该数据库源节点。
- ▶ 附加要列出其属性的 Microsoft 数据库建模节点。
- ▶ 打开该数据库建模节点。
- ▶ 选择专家选项卡。

此时会显示该节点的可用 `msas_parameters` 属性。

图片 18-1
算法参数显示示例



Microsoft 模型块属性

下列属性用于使用 Microsoft 数据库建模节点创建的模型块。

MS 决策树

aplymstreenode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。
datasource	string	SQL Server ODBC 数据源名 (DSN) 的名称。
sql_generate	flag	启用 SQL 生成。

MS 线性回归

aplymsregressionnode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。

MS 神经网络

appliesneuralnetworknode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。

MS Logistic 回归

applieslogisticnode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。

MS 时间序列

appliestimeseriesnode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。
start_from	new_prediction historical_prediction	指定是进行未来预测还是历史预测。
new_step	number	定义未来预测的开始时间周期。
historical_step	number	定义历史预测的开始时间周期。
end_step	number	定义预测的结束时间周期。

MS 序列聚类

appliessequenceclusternode 属性	值	描述
analysis_database_name	string	可以直接在流中对此节点进行评分。此属性用于标识 Analysis Services 数据库的名称。
analysis_server_name	string	分析服务器主机的名称。

Oracle 建模的节点属性

Oracle 建模节点属性

以下属性通用于 Oracle 数据库建模节点。

通用 Oracle 节点属性	值	属性说明
target	field	
inputs	字段列表	
partition	field	用于将数据区分为用于模型构建的训练、测试和验证阶段的各种样本的字段。
datasource		
username		
password		
epassword		
use_model_name	flag	
model_name	string	新模型的自定义名称。
use_partitioned_data	flag	如果定义了分区字段，则此选项可确保仅训练分区的数据用于构建模型。
unique_field	field	
auto_data_prep	flag	启用或禁用 Oracle 自动数据准备功能（仅 11g 数据库）。
costs	structured	形式如下的结构化属性： [<code>{drugA drugB 1.5} {drugA drugC 2.1}</code>], { } 中的参数为实际预测成本。
mode	Simple Expert	如在各个节点属性中注释的那样，如果设置为 Simple，会导致忽略某些属性。
use_prediction_probability	flag	
prediction_probability	string	
use_prediction_set	flag	

Oracle Naive Bayes

以下属性可用于类型为 oranbnode 的节点。

oranbnode 属性	值	属性说明
singleton_threshold	number	0.0 - 1.0.*
pairwise_threshold	number	0.0 - 1.0.*
priors	Data Equal Custom	
custom_priors	structured	形式如下的结构化属性： set :oranbnode.custom_priors = [<code>{drugA 1} {drugB 2} {drugC 3} {drugX 4} {drugY 5}</code>]

* 如果 mode 设置为 Simple，则忽略属性。

Oracle Adaptive Bayes

以下属性可用于类型为 `oraabnode` 的节点。

oraabnode 属性	值	属性说明
<code>model_type</code>	SingleFeature MultiFeature NaiveBayes	
<code>use_execution_time_limit</code>	flag	*
<code>execution_time_limit</code>	integer	值必须大于 0。*
<code>max_naive_bayes_predictors</code>	integer	值必须大于 0。*
<code>max_predictors</code>	integer	值必须大于 0。*
<code>priors</code>	Data Equal Custom	
<code>custom_priors</code>	structured	形式如下的结构化属性： set :oraabnode.custom_priors = [{drugA 1} {drugB 2} {drugC 3} {drugX 4} {drugY 5}]

* 如果 `mode` 设置为 `Simple`，则忽略属性。

Oracle Support Vector Machines

以下属性可用于类型为 `orasvmnode` 的节点。

orasvmnode 属性	值	属性说明
<code>active_learning</code>	Enable Disable	
<code>kernel_function</code>	Linear Gaussian System	
<code>normalization_method</code>	zscore minmax none	
<code>kernel_cache_size</code>	integer	仅限 Gaussian kernel。值必须大于 0。*
<code>convergence_tolerance</code>	number	值必须大于 0。*
<code>use_standard_deviation</code>	flag	仅限 Gaussian kernel。*
<code>standard_deviation</code>	number	值必须大于 0。*
<code>use_epsilon</code>	flag	仅限回归模型。*
<code>epsilon</code>	number	值必须大于 0。*
<code>use_complexity_factor</code>	flag	*
<code>complexity_factor</code>	number	*
<code>use_outlier_rate</code>	flag	仅限单类变量。*
<code>outlier_rate</code>	number	仅限单类变量。 0.0 - 1.0。*

orasvmnode 属性	值	属性说明
weights	Data Equal Custom	
custom_weights	structured	形式如下的结构化属性： set :orasvmnode.custom_weights = [{drugA 1} {drugB 2} {drugC 3} {drugX 4} {drugY 5}]

* 如果 mode 设置为 Simple，则忽略属性。

Oracle 广义线性模型

以下属性可用于类型为 oraglmnode 的节点。

oraglmnode 属性	值	属性说明
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWith- Mean UseCom- pleteRecords	
use_row_weights	flag	*
row_weights_field	field	*
save_row_diagnostics	flag	*
row_diagnostics_table	string	*
coefficient_confidence	number	*
use_reference_category	flag	*
reference_category	string	*
ridge_regression	Auto Off On	*
parameter_value	number	*
vif_for_ridge	flag	*

* 如果 mode 设置为 Simple，则忽略属性。

Oracle 决策树

以下属性可用于类型为 oradecisiontreenode 的节点。

oradecisiontreenode 属性	值	属性说明
use_costs	flag	
impurity_metric	Entropy Gini	
term_max_depth	integer	2 - 20.*
term_minpct_node	number	0.0 - 10.0.*
term_minpct_split	number	0.0 - 20.0.*

oradecisiontreenode 属性	值	属性说明
term_minrec_node	integer	值必须大于 0。*
term_minrec_split	integer	值必须大于 0。*
display_rule_ids	flag	*

* 如果 mode 设置为 Simple，则忽略属性。

Oracle O-Cluster

以下属性可用于类型为 oraoclusternode 的节点。

oraoclusternode 属性	值	属性说明
max_num_clusters	integer	值必须大于 0。
max_buffer	integer	值必须大于 0。*
sensitivity	number	0.0 - 1.0.*

* 如果 mode 设置为 Simple，则忽略属性。

Oracle KMeans

以下属性可用于类型为 orakmeansnode 的节点。

orakmeansnode 属性	值	属性说明
num_clusters	integer	值必须大于 0。
normalization_method	zscore minmax none	
distance_function	Euclidean Cosine	
iterations	integer	0 - 20.*
conv_tolerance	number	0.0 - 0.5.*
split_criterion	Variance Size	默认值为 Variance。*
num_bins	integer	值必须大于 0。*
block_growth	integer	1 - 5.*
min_pct_attr_support	number	0.0 - 1.0.*

* 如果 mode 设置为 Simple，则忽略属性。

Oracle NMF

以下属性可用于类型为 oranmfnode 的节点。

oranmfnode 属性	值	属性说明
normalization_method	minmax none	
use_num_features	flag	*

oramfnode 属性	值	属性说明
num_features	integer	0 - 1, 默认值由算法根据数据估计得出。*
random_seed	number	*
num_iterations	integer	0 - 500.*
conv_tolerance	number	0.0 - 0.5.*
display_all_features	flag	*

* 如果 mode 设置为 Simple, 则忽略属性。

Oracle Apriori

以下属性可用于类型为 oraapriorinode 的节点。

oraapriorinode 属性	值	属性说明
content_field	field	
id_field	field	
max_rule_length	integer	2 - 20.
min_confidence	number	0.0 - 1.0.
min_support	number	0.0 - 1.0.
use_transactional_data	flag	

Oracle 最小描述符长度 (MDL)

没有为 oramdlnode 类型的节点定义具体属性。请参阅本章节开头部分的通用 Oracle 属性。

Oracle 属性重要性 (AI)

以下属性可用于类型为 oraainode 的节点。

oraainode 属性	值	属性说明
custom_fields	flag	如果为 true, 则允许您指定当前节点的目标、输入和其他字段。如果为 false, 则使用来自上游类型节点的当前设置。
selection_mode	ImportanceLevel Importance-Value TopN	
select_important	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“重要”字段。
important_label	string	指定“重要”级别的标签。
select_marginal	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“边际”字段。
marginal_label	string	指定“边际”级别的标签。
important_above	number	0.0 - 1.0.
select_unimportant	flag	在 selection_mode 设置为 ImportanceLevel 时, 指定是否选择“不重要”字段。

oraainode 属性	值	属性说明
unimportant_label	string	指定“不重要”级别的标签。
unimportant_below	number	0.0 - 1.0.
importance_value	number	在 selection_mode 设置为 ImportanceValue 时，指定要使用的截断值。接受从 0 到 100 之间的值。
top_n	number	在 selection_mode 设置为 TopN 时，指定要使用的截断值。接受从 0 到 1000 之间的值。

Oracle 模型块属性

下列属性用于使用 Oracle 模型创建的模型块。

Oracle Naive Bayes

没有为 applyoranbnode 类型的节点定义具体属性。

Oracle Adaptive Bayes

没有为 applyoraabnode 类型的节点定义具体属性。

Oracle Support Vector Machines

没有为 applyorasvmnode 类型的节点定义具体属性。

Oracle 决策树

以下属性可用于类型为 applyoradecisiontreenode 的节点。

applyoradecisiontreenode 属性	值	属性说明
use_costs	flag	
display_rule_ids	flag	

Oracle O-Cluster

没有为 applyoraoclusternode 类型的节点定义具体属性。

Oracle KMeans

没有为 applyorakmeansnode 类型的节点定义具体属性。

Oracle NMF

下列属性用于 `applyoranmfnode` 类型的节点：

applyoranmfnode 属性	值	属性说明
<code>display_all_features</code>	flag	

Oracle Apriori

该模型块不能应用于脚本。

Oracle MDL

该模型块不能应用于脚本。

IBM DB2 建模的节点属性**IBM DB2 建模节点属性**

以下属性通用于 IBM InfoSphere Warehouse (ISW) 数据库建模节点。

通用 ISW 节点属性	值	属性说明
<code>inputs</code>	字段列表	
<code>datasource</code>		
<code>username</code>		
<code>password</code>		
<code>epassword</code>		
<code>enable_power_options</code>	flag	
<code>power_options_max_memory</code>	integer	值必须大于 32。
<code>power_options_cmdline</code>	string	
<code>mining_data_custom_sql</code>	string	
<code>logical_data_custom_sql</code>	string	
<code>mining_settings_custom_sql</code>		

ISW 决策树

以下属性可用于类型为 `db2imtreenode` 的节点。

db2imtreenode 属性	值	属性说明
<code>target</code>	field	
<code>perform_test_run</code>	flag	
<code>use_max_tree_depth</code>	flag	
<code>max_tree_depth</code>	integer	大于 0 的值。
<code>use_maximum_purity</code>	flag	
<code>maximum_purity</code>	number	介于 0 到 100 之间的数字。

db2imtreenode 属性	值	属性说明
use_minimum_internal_cases	flag	
minimum_internal_cases	integer	大于 1 的值。
use_costs	flag	
costs	structured	形式如下的结构化属性： [[{drugA drugB 1.5} {drugA drugC 2.1}], {}] 中的参数为实际预测成本。

ISW 关联

以下属性可用于类型为 db2imassocnode 的节点。

db2imassocnode 属性	值	属性说明
use_transactional_data	flag	
id_field	field	
content_field	field	
data_table_layout	basic lim- ited_length	
max_rule_size	integer	值必须大于 2。
min_rule_support	number	0 - 100%
min_rule_confidence	number	0 - 100%
use_item_constraints	flag	
item_constraints_type	Include Exclude	
use_taxonomy	flag	
taxonomy_table_name	string	存储分类法详细信息的 DB2 表的名称。
taxonomy_child_column_name	string	分类法表中子列的名称。该子列包含项目名或类别名。
taxonomy_parent_column_name	string	分类法表中父列的名称。该父列包含类别名。
load_taxonomy_to_table	flag	控制是否应在构建模型时将 IBM® SPSS® Modeler 中存储的分类法信息上载至分类法表。请注意，如果分类法表已经存在，则会将其丢弃。分类法信息存储在模型构建节点中，可以使用编辑类别和编辑分类法按钮进行编辑。

ISW 序列

以下属性可用于类型为 db2imsequencenode 的节点。

db2imsequencenode 属性	值	属性说明
id_field	field	
group_field	field	
content_field	field	
max_rule_size	integer	值必须大于 2。
min_rule_support	number	0 - 100%

db2imsequencenode 属性	值	属性说明
min_rule_confidence	number	0 - 100%
use_item_constraints	flag	
item_constraints_type	Include Exclude	
use_taxonomy	flag	
taxonomy_table_name	string	存储分类法详细信息的 DB2 表的名称。
taxonomy_child_column_name	string	分类法表中子列的名称。该子列包含项目名或类别名。
taxonomy_parent_column_name	string	分类法表中父列的名称。该父列包含类别名。
load_taxonomy_to_table	flag	控制是否应在构建模型时将 SPSS Modeler 中存储的分类法信息上载至分类法表。请注意，如果分类法表已经存在，则会将其丢弃。分类法信息存储在模型构建节点中，可以使用编辑类别和编辑分类法按钮进行编辑。

ISW 回归

以下属性可用于类型为 db2imregnode 的节点。

db2imregnode 属性	值	属性说明
target	field	
regression_method	transform linear polynomial rbf	参阅下一表格以获得当 regression_method 被设为 rbf 时适用的属性。
perform_test_run	field	
limit_rsquared_value	flag	
max_rsquared_value	number	介于 0.0 到 1.0 之间的值。
use_execution_time_limit	flag	
execution_time_limit_mins	integer	大于 0 的值。
use_max_degree_polynomial	flag	
max_degree_polynomial	integer	
use_intercept	flag	
use_auto_feature_selection_method	flag	
auto_feature_selection_method	normal adjusted	
use_min_significance_level	flag	
min_significance_level	number	
use_min_significance_level	flag	

以下属性仅在 regression_method 被设为 rbf 时适用。

db2imregnode 属性	值	属性说明
use_output_sample_size	flag	如果真，自动将值设置为默认。

output_sample_size	integer	默认值是 2。 最小值为 1。
use_input_sample_size	flag	如果真，自动将值设置为默认。
input_sample_size	integer	默认值是 2。 最小值为 1。
use_max_num_centers	flag	如果真，自动将值设置为默认。
max_num_centers	integer	默认值是 20。 最小值为 1。
use_min_region_size	flag	如果真，自动将值设置为默认。
min_region_size	integer	默认值是 15。 最小值为 1。
use_max_data_passes	flag	如果真，自动将值设置为默认。
max_data_passes	integer	默认值是 5。 最小值为 2。
use_min_data_passes	flag	如果真，自动将值设置为默认。
min_data_passes	integer	默认值是 5。 最小值为 2。

ISW 聚类

以下属性可用于类型为 `db2imclusternode` 的节点。

db2imclusternode 属性	值	属性说明
cluster_method	demographic kohonen birch	
kohonen_num_rows	integer	
kohonen_num_columns	integer	
kohonen_passes	integer	
use_num_passes_limit	flag	
use_num_clusters_limit	flag	
max_num_clusters	integer	大于 1 的值。
birch_dist_measure	log_likeli- hood euclidean	默认值是 log_likelihood。
birch_num_cfleaves	integer	默认值是 1000。
birch_num_refine_passes	integer	默认值是 3；最小值是 1。
use_execution_time_limit	flag	
execution_time_limit_mins	integer	大于 0 的值。
min_data_percentage	number	0 - 100%
use_similarity_threshold	flag	
similarity_threshold	number	介于 0.0 到 1.0 之间的值。

ISW Naive Bayes

以下属性可用于类型为 `db2imnbsnode` 的节点。

db2imnbsnode 属性	值	属性说明
<code>perform_test_run</code>	flag	
<code>probability_threshold</code>	number	默认值是 .001。 最小值为 0；最大值为 1.000
<code>use_costs</code>	flag	
<code>costs</code>	structured	形式如下的结构化属性： [[drugA drugB 1.5] {drugA drugC 2.1}], {} 中的参数为实际预测成本。

ISW Logistic 回归

以下属性可用于类型为 `db2imlognode` 的节点。

db2imlognode 属性	值	属性说明
<code>perform_test_run</code>	flag	
<code>use_costs</code>	flag	
<code>costs</code>	structured	形式如下的结构化属性： [[drugA drugB 1.5] {drugA drugC 2.1}], {} 中的参数为实际预测成本。

ISW 时间序列

注意：输入字段参数没有用于此节点。如果在脚本中找到输入字段参数，则显示一个警告，说明该节点有时间和目标作为输入字段，但没有输入字段。

以下属性可用于类型为 `db2imtimeseriesnode` 的节点。

db2imtimeseriesnode 属性	值	属性说明
<code>time</code>	field	允许整数、时间或日期。
<code>targets</code>	字段列表	
<code>forecasting_algorithm</code>	arima exponen- tial_smooth- ing sea- sonal_trend_de- composition	
<code>forecasting_end_time</code>	auto integer date time	
<code>use_records_all</code>	boolean	如果假，必须设置 <code>use_records_start</code> 和 <code>use_records_end</code> 。
<code>use_records_start</code>	integer / time / date	取决于时间字段的类型

db2imtimeseriesnode 属性	值	属性说明
use_records_end	integer / time / date	取决于时间字段的类型
interpolation_method	none linear exponential_splines cubic_splines	

IBM DB2 模型块属性

下列属性用于使用 IBM DB2 ISW 模型创建的模型块。

ISW 决策树

没有为 applydb2imtreenode 类型的节点定义具体属性。

ISW 关联

该模型块不能应用于脚本。

ISW 序列

该模型块不能应用于脚本。

ISW 回归

没有为 applydb2imregnode 类型的节点定义具体属性。

ISW 聚类

没有为 applydb2imclusternode 类型的节点定义具体属性。

ISW Naive Bayes

没有为 applydb2imnbnode 类型的节点定义具体属性。

ISW Logistic 回归

没有为 applydb2imlognode 类型的节点定义具体属性。

ISW 时间序列

该模型块不能应用于脚本。

IBM Netezza Analytics 建模节点属性

Netezza 建模节点属性

以下属性通用用于 IBM Netezza 数据库建模节点。

通用 Netezza 节点属性	值	属性说明
custom_fields	flag	如果为 true, 则允许您指定当前节点的目标、输入和其他字段。如果为 false, 则使用来自上游类型节点的当前设置。
inputs	[field1 ... fieldN]	模型使用的输入或预测变量字段。
target	field	目标字段 (连续或分类)。
record_id	field	要作为唯一记录标识符使用的字段。
use_upstream_connection	flag	如果为 true (默认), 则使用上游节点中指定的连接详细信息。如果指定了 move_data_to_connection, 则不使用。
move_data_connection	flag	如果为 true, 则将数据移动到由 connection 指定的数据库。如果指定了 use_upstream_connection, 则不使用。
connection	structured	用于存储模型的 Netezza 数据库的连接字符串。形式如下的结构化属性: ['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' {true false}] 其中: <dsn> 是数据源名称 <username> 与 <psw> 是数据库的用户名和密码 <catname> 是目录名称 <conn_attribs> 是连接属性 true false 表示是否需要密码。
table_name	string	要存储模型数据库表的名称。
use_model_name	flag	如果为 true, 使用由 model_name 指定的名称作为模型名称, 否则采用系统创建的模型名称。
model_name	string	新模型的自定义名称。
include_input_fields	flag	如果为 true, 向下游传递所有输入字段, 否则仅传递模型产生的 record_id 和字段。

Netezza 决策树

以下属性可用于类型为 netezzadectreenode 的节点。

netezzadectreenode 属性	值	属性说明
impurity_measure	Entropy Gini	杂质测量, 用于评估分割树的最佳位置。
max_tree_depth	integer	树可以增长的最大级数。默认值是 62 (最大允许值)。
min_improvement_splits	number	出现分割杂质的最小改进。默认值是 .01。

netezzadectreenode 属性	值	属性说明
min_instances_split	integer	在可以分割前还未分割记录的最小数量。默认值是 2（最小允许值）。
weights	structured	类的相对权重。形式如下的结构化属性： set :netezza_dectree.weights = [[drugA 0.3] [drugB 0.6]] 对于所有类，权重的默认值是 1。
pruning_measure	Acc wAcc	默认值是 Acc（精确度）。如果要在应用修剪时将类权重考虑在内，可使用 wAcc（加权精确度）替代。
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	默认情况下，使用 allTrainingData 来估计模型精确度。使用 partitionTrainingData 来指定要使用训练数据的百分比，或 useOtherTable 来使用源自指定数据库表的训练数据集。
perc_training_data	number	如果 prune_tree_options 被设为 partitionTrainingData，指定要用于训练的数据百分比。
prune_seed	integer	当 prune_tree_options 被设为 partitionTrainingData 时，用于重复分析结果的随机种子，默认值是 1。
pruning_table	string	用于估计模型精确度的单独修剪数据集的表名称。
compute_probabilities	flag	如果为 true，产生置信水平（概率）字段和预测字段。

Netezza K-Means

以下属性可用于类型为 netezzakmeansnode 的节点。

netezzakmeansnode 属性	值	属性说明
distance_measure	Euclidean Manhattan Canberra maximum	用于测量数据点之间距离的方法。
num_clusters	integer	要创建的聚类数量，默认值是 3。
max_iterations	integer	停止模型训练的算法迭代数量，默认值是 5。
rand_seed	integer	用于重复分析结果的随机种子，默认值是 12345。

Netezza 贝叶斯网络

以下属性可用于类型为 netezزابayesnode 的节点。

netezزابayesnode 属性	值	属性说明
base_index	integer	分配给第一个输出字段以方便内部管理的数字标识符，默认值是 777。
sample_size	integer	当属性数量过多时，要采用的样本大小，默认值是 10,000。

netezabayesnode 属性	值	属性说明
display_additional_information	flag	如果为 true, 在消息对话框中显示更多进度信息。
type_of_prediction	best neighbors nn-neighbors	要使用的预测算法类型。最佳 (最相关近邻元素)、近邻元素 (近邻元素的加权预测) 或 nn-近邻元素 (非 Null 近邻元素)。

Netezza Naive Bayes

以下属性可用于类型为 netezanaivebayesnode 的节点。

netezanaivebayesnode 属性	值	属性说明
compute_probabilities	flag	如果为 true, 产生置信水平 (概率) 字段和预测字段。
use_m_estimation	flag	如果为 true, 使用 m 估计技术, 以避免在估计期间出现零概率。

Netezza KNN

以下属性可用于类型为 netezaknnnode 的节点。

netezaknnnode 属性	值	属性说明
weights	structured	用于为各个类分配权重的结构化属性。示例: set :netezaknnnode.weights = [{drugA 0.3} {drugB 0.6}]
distance_measure	Euclidean Manhattan Canberra Maximum	用于测量数据点之间距离的方法。
num_nearest_neighbors	integer	特定个案的最近相邻元素数量, 默认值是 3。
standardize_measurements	flag	如果为 true, 将标准化连续输入字段的测量结果, 然后再计算距离值。
use_coresets	flag	如果为 true, 将针对大型数据集采用核心集抽样以加快计算过程。

Netezza 分裂式聚类

以下属性可用于类型为 netezadivclusternode 的节点。

netezadivclusternode 属性	值	属性说明
distance_measure	Euclidean Manhattan Canberra Maximum	用于测量数据点之间距离的方法。
max_iterations	integer	停止模型训练之前的算法迭代最大次数, 默认值是 5。
max_tree_depth	integer	数据集可以细分的最大层级数, 默认值是 3。
rand_seed	integer	用于重复分析的随机种子, 默认值是 12345。

netezzadivclusternode 属性	值	属性说明
min_instances_split	integer	可分割的最小记录数，默认值是 5。
level	integer	要评分的记录层次层级，默认值是 -1。

Netezza PCA

以下属性可用于类型为 netezzapcanode 的节点。

netezzapcanode 属性	值	属性说明
center_data	flag	如果为 true (默认)，将在分析之前执行数据中心化（也称为“平均值消去法”）。
perform_data_scaling	flag	如果为 true，将在分析之前执行数据换算。在不同变量采用不同单位时，这样可以降低分析的任意性。
force_eigensolve	flag	如果为 true，使用低准确度但快速的方法来寻找主成分。
pc_number	integer	要减小的数据集的主成分个数，默认值是 1。

Netezza 回归树

以下属性可用于类型为 netezzaregtreenode 的节点。

netezzaregtreenode 属性	值	属性说明
max_tree_depth	integer	在根节点下树可以增长的最大级数，默认值是 10。
split_evaluation_measure	Variance	类杂质测量，用于评估分割树的最佳位置，默认值（当前唯一选项）是 Variance。
min_improvement_splits	number	在树中创建新分割之前要减少的最小杂质质量。
min_instances_split	integer	可分割的最小记录数。
pruning_measure	mse r2 pearson spearman	要使用的修剪方法
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	默认情况下，使用 allTrainingData 来估计模型精确度。使用 partitionTrainingData 来指定要使用训练数据的百分比，或 useOtherTable 来使用源自指定数据库表的训练数据集。
perc_training_data	number	如果 prune_tree_options 被设为 PercTrainingData，指定要用于训练的数据百分比。
prune_seed	integer	当 prune_tree_options 被设为 PercTrainingData 时，用于重复分析结果的随机种子，默认值是 1。
pruning_table	string	用于估计模型精确度的单独修剪数据集的表名称。
compute_probabilities	flag	如果为 true，指定应在输出中包含所分配类的方差。

Netezza 线性回归

以下属性可用于类型为 `netezzalineressionnode` 的节点。

netezzalineressionnode 属性	值	属性说明
<code>use_svd</code>	flag	如果为 true, 使用奇异值分解矩阵而不是原始矩阵, 以加快计算过程和提高数字准确性。
<code>include_intercept</code>	flag	如果为 true (默认), 可提高解的总体准确性。
<code>calculate_model_diagnostics</code>	flag	如果为 true, 在模型上计算诊断数据。

Netezza 时间序列

以下属性可用于类型为 `netezzatimeseriesnode` 的节点。

netezzatimeseriesnode 属性	值	属性说明
<code>time_points</code>	field	输入字段包含时间序列的日期或时间值。
<code>time_series_ids</code>	field	包含时间序列 ID 的输入字段。如果输入包含一个以上的时间序列, 则使用这个字段。
<code>model_table</code>	field	存储 Netezza 时间序列模型的数据库表名称。
<code>description_table</code>	field	包含时间序列名称和说明的输入表名称。
<code>seasonal_adjustment_table</code>	field	存储由指数平滑或季节趋势分解算法计算的按季节调整值的输出表名称。
<code>algorithm_name</code>	SpectralAnalysis 或 spectral ExponentialSmoothing 或 smoothing ARIMA Seasonal-TrendDecomposition 或 std	用于时间序列建模的算法。
<code>trend_name</code>	string	指数平滑的趋势类型。
<code>seasonality_type</code>	string	指数平滑的季节类型。
<code>interpolation_method</code>	linear cubic spline exponential- spline	要使用的插值方法。
<code>earliest_time</code>	field	如果使用时间序列的一部分, 则为开始时间。
<code>latest_time</code>	field	如果使用时间序列的一部分, 则为结束时间。
<code>p</code>	integer	ARIMA - 非季节的自相关度。
<code>q</code>	integer	ARIMA - 非季节的派生值。
<code>d</code>	integer	ARIMA - 模型中非季节移动平均阶数的值。
<code>sp</code>	integer	ARIMA - 季节的自相关度。
<code>sq</code>	integer	ARIMA - 季节的派生值。

netezzatimeseriesnode 属性	值	属性说明
sd	integer	ARIMA - 模型中季节移动平均阶数的值。
period	integer	季节周期的长度，与 units_period 共同指定。不适用于光谱分析。
units_period	Milliseconds Seconds Minutes Hours Days Weeks Quarters Years	表示 period 使用的单位。例如，对于每周时间序列，为 period 使用 1，为 units_period 使用 Weeks。
include_history	flag	指明是否在输出中包含历史值。
include_interpolated_values	flag	指明是否在输出中包含插补的值。如果 include_history 为 false，则不适用。

Netezza 广义线性

以下属性可用于类型为 netezzaglmnode 的节点。

netezzaglmnode 属性	值	属性说明
response_variable_distribution	bernoulli gaussian poisson binomial negativebinomial wald gamma	分布类型；默认为 bernoulli。
distribution_parameter	number	要使用的分布参数值。仅于 distribution 为 Negativebinomial 时适用。
trials	integer	仅于 distribution 为 Binomial 时适用。当目标响应为发生在一组试验中的事件数时，target 字段包含事件数，trials 字段包含试验数。
model_table	field	存储 Netezza 广义线性模型的数据库表名称。
max_iterations	integer	算法进行迭代的最多次数；默认为 20 次。
max_error	number	最大误差值（以科学计数法），达到此值后，算法停止寻找最佳拟合模型。默认值为 -3，表示 1E-3 或 0.001。
error_threshold	number	数值（用科学计数法），低于此值的所有误差均被视为 0 值。默认值为 -7，表示误差值若低于 1E-7（或 0.000001），则被视为不显著。

netezzaglmnode 属性	值	属性说明
link_function	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	要使用的联接函数；默认为 logit。
link_function_parameter	number	要使用的联接函数参数值。仅于 link_function 为 power 或 oddspower 时适用。
intercept	flag	如果 true，则在模型中包含截距。

Netezza 模型块属性

以下属性通用用于 Netezza 数据库模型块。

通用 Netezza 模型块属性	值	属性说明
connection	string	用于存储模型的 Netezza 数据库的连接字符串。
table_name	string	存储模型数据库表的名称。

其他模型块属性与对应建模节点的属性相同。

模型块的脚本名称如下。

模型块	脚本名称
决策树	applynetezzadectreenode
K-Means	applynetezzakmeansnode
贝叶斯网络	applynetezزابayesnode
Naive Bayes	applynetezzanaivebayesnode
KNN	applynetezzaknnnode
分裂式聚类	applynetezzagdivclusternode
主成分分析 (PCA)	applynetezzapcanode
回归树	applynetezzaregtreenode
线性回归	applynetezzalineressionnode
时间序列	applynetezzatimeseriesnode
广义线性	applynetezzaglmnode

输出节点属性

输出节点的属性与其他类型节点的属性略有不同。输出节点属性不是指特定的节点选项，而是存储对输出对象的引用。这在从表中提取值并将其设置为流参数时非常有用。本章节说明输出节点的可用的脚本属性。

analysisnode 属性



“分析”节点评估预测模型生成准确预测的能力。“分析”节点执行一个或多个模型块的预测值和实际值之间的各种比较。“分析”节点也可以对比各个预测模型。有关详细信息，请参阅第 6 章中的分析节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

analysisnode属性	数据类型	属性说明
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。
output_format	Text (.txt) HTML (.html) Output (.cou)	用于指定输入的类型。
by_fields	[field field field]	

analysisnode属性	数据类型	属性说明
full_filename	string	如果是磁盘、数据或 HTML 输出，则此属性指输出文件的名称。
coincidence	flag	
performance	flag	
confidence	flag	
threshold	number	
improve_accuracy	number	
inc_user_measure	flag	
user_if	expr	
user_then	expr	
user_else	expr	
user_compute	[Mean Sum Min Max SDev]	

dataauditnode 属性



数据审核节点将首先全面检查数据，这些数据包括每个字段的汇总统计量、直方图和分布以及有关离群值、缺失值和极值的信息。结果显示在易于读取的矩阵中，该矩阵可以排序并且可以用于生成完整大小的图表和数据准备节点。有关详细信息，请参阅第 6 章中的数据审核节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

dataauditnode属性	数据类型	属性说明
custom_fields	flag	
fields	[field1 ... fieldN]	
overlay	field	
display_graphs	flag	用于打开或关闭输出矩阵中图形的显示。
basic_stats	flag	

dataauditnode属性	数据类型	属性说明
advanced_stats	flag	
median_stats	flag	
calculate	Count Breakdown	用于计算缺失值。选择两种计算方法中的一种、两种，或均不选择。
outlier_detection_method	std iqr	用于指定离群值和极值的检测方法。
outlier_detection_std_outlier	number	如果 outlier_detection_method 是 std, 则指定用于定义离群值的数值。
outlier_detection_std_extreme	number	如果 outlier_detection_method 是 std, 则指定用于定义极值的数值。
outlier_detection_iqr_outlier	number	如果 outlier_detection_method 是 iqr, 则指定用于定义离群值的数值。
outlier_detection_iqr_extreme	number	如果 outlier_detection_method 是 iqr, 则指定用于定义极值的数值。
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真, 则指定使用的名称。
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	Formatted ... /Tabs Delimited (.csv) HTML (.html) Output (.cou)	用于指定输入的类型。
paginate_output	flag	当 output_format 是 HTML 时, 使输出分页。
lines_per_page	number	与 paginate_output 一起使用时, 指定每个输出页中的行数。
full_filename	string	

matrixnode 属性



矩阵节点将创建一个字段关系表。此节点最常用于显示两个符号字段间的关系，但也可用于显示标志字段或数字字段间的关系。有关详细信息，请参阅第 6 章中的矩阵节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create matrixnode
# "Settings" tab
set :matrixnode.fields = Numerics
```

```

set :matrixnode.row = 'K'
set :matrixnode.column = 'Na'
set :matrixnode.cell_contents = Function
set :matrixnode.function_field = 'Age'
set :matrixnode.function = Sum
# "Appearance" tab
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
# "Output" tab
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

```

matrixnode属性	数据类型	属性说明
fields	Selected Flags Numerics	
row	field	
column	field	
include_missing_values	flag	指定在行和列输出中是否包含用户缺失值（空白）和系统缺失值（Null 值）。
cell_contents	CrossTabs Function	
function_field	string	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	number	如果非零，则为真。
highlight_bottom	number	如果非零，则为真。
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	flag	
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。

matrixnode属性	数据类型	属性说明
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	Formatted (. tab) Delimited (. csv) HTML (. html) Output (. cou)	用于指定输入的类型。 Formatted 和 Delimited 格式都可使用限定符 transposed, 此符号可转置表中的行和列; 例如: NODE.output_format=transposed Delimited
paginate_output	flag	当 output_format 是 HTML 时, 使输出分页。
lines_per_page	number	与 paginate_output 一起使用时, 指定每个输出页中的行数。
full_filename	string	

meansnode 属性



平均值节点在独立组之间或相关字段对之间进行平均值比较以检验是否存在显著差别。例如, 您可以比较开展促销前后的平均收入, 或者将来自未接受促销客户的收入与接受促销客户的收入进行比较。有关详细信息, 请参阅第 6 章中的平均值节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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"
```

meansnode属性	数据类型	属性说明
means_mode	BetweenGroups BetweenFields	指定要在数据上执行的均值统计量的类型。
test_fields	[field1 ... fieldn]	指定当 means_mode 设置为 BetweenGroups 时的检验字段。
grouping_field	field	指定分组字段。
paired_fields	[{field1 field2} {field3 field4} ...]	指定当 means_mode 设置为 BetweenFields 时要使用的字段对。
label_correlations	flag	指定在输出中是否显示相关标签。仅在当 means_mode 设置为 BetweenFields 时才应用此设置。

meansnode属性	数据类型	属性说明
correlation_mode	Probability Absolute	指定用概率还是绝对值标注相关。
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	number	当 correlation_mode 设置为 Probability 时, 指定弱相关的截断值。这必须是 0 到 1 之间的一个值—例如, 0.90。
strong_above_probability	number	强相关的截断值。
weak_below_absolute	number	当 correlation_mode 设置为 Absolute 时, 指定弱相关的截断值。这必须是 0 到 1 之间的一个值—例如, 0.90。
strong_above_absolute	number	强相关的截断值。
unimportant_label	string	
marginal_label	string	
important_label	string	
unimportant_below	number	低字段重要性的截断值。这必须是 0 到 1 之间的一个值—例如, 0.90。
important_above	number	
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	使用的名称。
output_mode	Screen File	指定从输出节点中生成的输出的目标位置。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	指定输出的类型。
full_filename	string	
output_view	Simple Advanced	指定在输出中显示简单视图还是高级视图。

reportnode 属性



报告节点可创建格式化报告, 其中包含固定文本、数据及得自数据的其他表达式。可使用文本模板指定报告的格式以定义固定文本和数据输出结构。通过使用模板中的 HTML 标记和在“输出”选项卡上设置选项, 可以提供自定义文本格式。通过使用模板中的 CLEM 表达式, 可以包括数据值和其他条件输出。有关详细信息, 请参阅第 6 章中的报告节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

reportnode属性	数据类型	属性说明
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	HTML (.html) Text (.txt) Output (.cou)	用于指定输入的类型。
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。
text	string	
full_filename	string	
highlights	flag	
title	string	
lines_per_page	number	

setglobalsnode 属性



设置全局节点扫描数据并计算可在 CLEM 表达式中使用的汇总值。例如，可以用该节点为一个名为年龄的字段计算统计量并通过插入函数 @GLOBAL_MEAN(age) 在 CLEM 表达式中使用年龄的总均值。有关详细信息，请参阅第 6 章中的设置全局节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

setglobalsnode属性	数据类型	属性说明
globals	[Sum Mean Min Max SDev]	结构属性，在其中，必须使用下面的语法引用要设置的字段： set :setglobalsnode.globals.Age = [Sum Mean Min Max SDev]
clear_first	flag	
show_preview	flag	

statisticsnode 属性



统计量节点可提供有关数值字段的基本汇总信息。它可计算单个字段以及字段间的相关性的汇总统计量。有关详细信息，请参阅第 6 章中的统计量节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

statisticsnode属性	数据类型	属性说明
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	Text (.txt) HTML (.html) Output (.cou)	用于指定输入的类型。
full_filename	string	
examine	[field field field]	

statisticsnode属性	数据类型	属性说明
correlate	[field field field]	
statistics	[Count Mean Sum Min Max Range Variance SDev SErr Median Mode]	
correlation_mode	Probability Absolute	指定用概率还是绝对值标注相关。
label_correlations	flag	
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	number	当 correlation_mode 设置为 Probability 时, 指定弱相关的截断值。这必须是 0 到 1 之间的一个值—例如, 0.90。
strong_above_probability	number	强相关的截断值。
weak_below_absolute	number	当 correlation_mode 设置为 Absolute 时, 指定弱相关的截断值。这必须是 0 到 1 之间的一个值—例如, 0.90。
strong_above_absolute	number	强相关的截断值。

statisticsoutputnode 属性



Statistics 输出节点可调用 IBM® SPSS® Statistics 过程以分析您的 IBM® SPSS® Modeler 数据。可以访问许多不同的 SPSS Statistics 分析过程。此节点需要 SPSS Statistics 的许可副本。有关详细信息, 请参阅第 8 章中的 Statistics 输出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

有关此节点属性的信息, 请参阅 [statisticsoutputnode 属性](#) 第 279 页码。

tablenode 属性



表节点以表格形式显示数据, 这些数据还可以写入到文件中。每当您需要检查数据值或将其导出为可轻松读取的形式时, 该节点则非常有用。有关详细信息, 请参阅第 6 章中的表节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

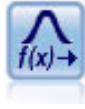
```
create tablenode
set :tablenode.highlight_expr = "Age > 30"
set :tablenode.output_format = HTML
set :tablenode.transpose_data = true
set :tablenode.full_filename = "C:/output/table_output.htm"
```

```
set :tablnode.paginate_output = true
set :tablnode.lines_per_page = 50
```

tablnode属性	数据类型	属性说明
full_filename	string	如果是磁盘、数据或 HTML 输出，则此属性指输出文件的名称。
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	用于指定输入的类型。
transpose_data	flag	导出前转置数据，使行表示字段，列表示记录。
paginate_output	flag	当 output_format 是 HTML 时，使输出分页。
lines_per_page	number	与 paginate_output 一起使用时，指定每个输出页中的行数。
highlight_expr	string	
output	string	只读属性，可保留对由节点构建的最后一个表的引用。
value_labels	[[Value LabelString] {Value LabelString} ...]	用于为值对指定标签。 例如， set :typenode.value_labels. 'Drug'=[[drugA label1] {drugB label2}]
display_places	integer	为字段设置显示的小数位数（仅用于以 REAL 存储的字段）。值为 -1 时将使用流默认值。 用法格式： NODE.display_places. FIELDNAME
export_places	integer	为字段设置导出时的小数位数（仅用于以 REAL 存储的字段）。值为 -1 时将使用流默认值。 用法格式： NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD COMMA	为字段设置十进制分隔符（仅用于以 REAL 存储的字段）。 用法格式： NODE.decimal_separator. FIELDNAME

tablenode属性	数据类型	属性说明
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	为字段设置日期格式（仅用于以 DATE 或 TIMESTAMP 存储的字段）。 用法格式： NODE. date_format. FIELDNAME 例如， 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"	为字段设置时间格式（仅用于以 TIME 或 TIMESTAMP 存储的字段）。 用法格式： NODE. time_format. FIELDNAME 例如， set :tablenode. time_format. set 'BOF_enter' = "HHMMSS"
column_width	integer	为字段设置列宽度。值为 -1 表示将列宽度设置为 Auto。 用法格式： NODE. column_width. FIELDNAME
justify	AUTO CENTER LEFT RIGHT	为字段设置列对齐格式。 用法格式： NODE. justify. FIELDNAME

transformnode 属性



通过变换节点可首先选择和以可视方式预览变换结果，然后再将其应用于选择的字段。有关详细信息，请参阅第 6 章中的变换节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create transformnode
set :transformnode.fields = [AGE INCOME]
set :transformnode.formula = Select
set :transformnode.formula_log_n = true
set :transformnode.formula_log_n_offset = 1
```

transformnode属性	数据类型	属性说明
fields	[字段1... 字段n]	要在变换中使用的字段。
formula	All Select	表示应计算所有变换还是选定的变换。
formula_inverse	flag	表示是否应使用逆变换。
formula_inverse_offset	number	表示公式中要使用的数据偏移量。除非用户指定，否则默认情况下设置为 0。
formula_log_n	flag	表示是否应使用 \log_n 变换。
formula_log_n_offset	number	
formula_log_10	flag	表示是否应使用 \log_{10} 变换。
formula_log_10_offset	number	
formula_exponential	flag	表示是否应使用指数 (e^x) 变换。
formula_square_root	flag	表示是否应使用平方根变换。
use_output_name	flag	指定是否使用自定义的输出名。
output_name	string	如果 use_output_name 为真，则指定使用的名称。
output_mode	Screen File	用于指定从输出节点中生成的输出的目标位置。
output_format	HTML (.html) Output (.cou)	用于指定输入的类型。
paginate_output	flag	当 output_format 是 HTML 时，使输出分页。
lines_per_page	number	与 paginate_output 一起使用时，指定每个输出页中的行数。
full_filename	string	表示要在文件输出中使用的文件名。

导出节点属性

通用导出节点属性

以下属性通用于所有导出节点。

属性	值	属性说明
publish_path	string	输入供发布的图像和参数文件使用的 rootname 名称。
publish_metadata	flag	指定是否生成介绍图像的输入和输出以及它们的数据模型的元数据文件。
publish_use_parameters	flag	指定是否在 *.par 文件中包含流参数。
publish_parameters	string list	指定要包括的参数。
execute_mode	export_data publish	指定是执行节点而不发布流，还是在执行节点时自动发布流。

cognosexportnode 属性



IBM Cognos BI 导出节点以 Cognos BI 数据库可读取的格式导出数据。有关详细信息，请参阅第 7 章中的 IBM Cognos BI 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

注意： 对于此节点，必须定义 Cognos 连接和 ODBC 连接。

Cognos 连接

Cognos 连接的属性如下。

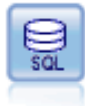
cognosexportnode 属性	数据类型	属性说明
cognos_connection	{ "field", "field", ..., "field" }	包含 Cognos 服务器连接详细信息的列表属性。格式为： { "Cognos_server_URL", login_mode, "namespace", "username", "password" } 其中： Cognos_server_URL 您要导出数据的目标 Cognos 服务器的 URL login_mode 表示是否使用匿名登录，其值为 true 或 false；如果设为 true，则应将以下字段设为 "" namespace 指定用于登录服务器的安全验证提供程序

cognosexportnode 属性	数据类型	属性说明
		username 和 password 为用于登录 Cognos 服务器的用户名和密码
cognos_package_name	string	您要将数据导出到的 Cognos 数据包的路径和名称, 例如: /Public Folders/MyPackage
cognos_datasource	string	
cognos_export_mode	Publish ExportFile	
cognos_filename	string	

ODBC 连接

ODBC 连接的属性和为下一节中的 `databaseexportnode` 列出的相同, 例外是 `datasource` 属性无效。

databaseexportnode 属性



数据库导出节点将数据写到与 ODBC 兼容的相关数据源。要写到 ODBC 数据源, 数据源必须存在且您必须拥有对数据源的写权限。有关详细信息, 请参阅第 7 章中的数据库导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

/*
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]
```

databaseexportnode 属性	数据类型	属性说明
datasource	string	
username	string	
password	string	
epassword	string	在执行过程中，此通道为只读。要生成加密密码，请使用“工具”菜单中的“密码工具”。有关详细信息，请参阅第 55 页码第 5 章中的生成加密密码。
table_name	string	
write_mode	Create Append Merge	
map	string	映射流字段名称到数据库列名称（只有当 write_mode 为 Merge 时才有效）。 示例： set :databaseexportnode.map.streamBP = 'databaseBP' 按字段位置支持多映射，例如： set :databaseexportnode.map=[{streamfield1 field1} {streamfield2 field2} {streamfield3 field3}] 对于合并，所有字段必须经过映射才能导出。数据库中不存在的字段名称将添加为新列。
key_fields	[field field ... field]	指定用作关键字的流字段；map 属性显示它在数据库中的对应项。
join	Database Add	示例： set : databaseexportnode.join = Database
drop_existing_table	flag	
delete_existing_rows	flag	
default_string_size	integer	
type		用于设置计划类型的结构属性。 用法格式： set :databaseexportnode.type.BP = 'VARCHAR(10)'
generate_import	flag	

databaseexportnode 属性	数据类型	属性说明
use_custom_create_table_command	flag	使用 custom_create_table 通道修改标准 CREATE TABLE SQL 命令。
custom_create_table_command	string	指定字符串命令代替标准 CREATE TABLE SQL 命令使用。
use_batch	flag	下列属性是数据库批量载入的高级选项。Use_batch 为真值时将关闭向数据库逐行提交的功能。
batch_size	number	指定在提交到内存前发送到数据库的记录数。
bulk_loading	Off ODBC External	指定批量载入的类型。下面列出了 ODBC 和 External 的其他选项。
odbc_binding	Row Column	指定通过 ODBC 批量载入时使用逐行绑定或逐列绑定。
loader_delimit_mode	Tab Space Other	对于通过外部程序的批量载入, 指定定界符的类型。选择 Other 连同 loader_other_delimiter 属性以指定定界符, 例如逗号 (,)。
loader_other_delimiter	string	
specify_data_file	flag	标志为真值时可激活下面的 data_file 属性, 在该属性中可以指定批量载入到数据库时写入的文件名和路径。
data_file	string	
specify_loader_program	flag	标志为真值时可激活下面的 loader_program 属性, 在该属性中可以指定外部载入程序脚本或程序的名称和位置。
loader_program	string	
gen_logfile	flag	标志为真值时可激活下面的 logfile_name, 在该属性中可以指定服务器上的文件的名称以生成错误日志。
logfile_name	string	
check_table_size	flag	标志为真值时允许进行表检查以确保数据库表大小的增加与从 IBM® SPSS® Modeler 导出的行数相符。
loader_options	string	指定载入程序的其他参数, 例如 -comment 和 -specialdir。
export_db_primarykey	flag	指定给定字段是否是主关键字。
use_custom_create_index_command	flag	如果标志为 true, 则为所有索引启用自定义 SQL。

databaseexportnode 属性	数据类型	属性说明
custom_create_index_command	string	当已启用自定义 SQL 时，指定用于创建索引的 SQL 命令。（创建特定索引时，该值可被覆盖，如下所示。）
indexes. INDEXNAME. fields		必要时创建指定的索引并列出将要包含在该索引中的字段名。
indexes. INDEXNAME. use_custom_create_index_command	flag	用于启用或禁用特定索引的自定义 SQL。
indexes. INDEXNAME. custom_create_command		指定用于指定索引的自定义 SQL。
indexes. INDEXNAME. remove	flag	如果标志为 true，则从索引集中删除指定的索引。
table_space	string	指定将要创建的表空格。
use_partition	flag	指定将使用分布散列字段。
partition_field	string	指定分布散列字段的内容。

注意：对于一些数据库，您可以指定创建的数据库表使用压缩导出（例如，相当于 SQL 中的 CREATE TABLE MYTABLE (...) COMPRESS YES;）。属性 use_compression 和 compression_mode 可供支持此功能，具体如下。

databaseexportnode 属性	数据类型	属性说明
use_compression	布尔逻辑	如果设置为 true，创建的表格使用压缩导出。
compression_mode	Row Page	为 SQL 服务器数据库设置压缩级别。
	Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low	为 Oracle 数据库设置压缩级别。请注意，需要至少 Oracle 11gR2 才可设置 OLTP、Query_High、Query_Low、Archive_High 和 Archive_Low 等值。

示例 - SQL 服务器

```
var DBSource
set DBSource = get node TestCompressionSQL
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Page

execute DBSource
```

示例 - Oracle 11gR1

```
var DBSource
set DBSource = get node TestCompressionOracle11gR1
```

```
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Direct_Load_Operations
```

```
execute DBSource
```

示例 - Oracle 11gR2

```
var DBSource
set DBSource = get node TestCompressionOracle11gR2
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Basic
```

```
execute DBSource
```

datacollectionexportnode 属性



IBM® SPSS® Data Collection 导出节点以 Data Collection 市场调查软件使用的格式输出数据。必须安装 Data Collection 数据库才可使用此节点。有关详细信息，请参阅第 7 章中的 IBM SPSS Data Collection 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

datacollectionexportnode 属性	数据类型	属性说明
metadata_file	string	要导出的元数据文件的名称。
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	flag	指定导出的 .mdd 文件是否应包括 Data Collection 系统变量。
casedata_file	string	要导出观测数据的 .sav 文件的名称。
generate_import	flag	

excelexportnode 属性



Excel 导出节点以 Microsoft Excel 格式 (.xls) 输出数据。也可以选择在执行节点时自动启动 Excel 并打开导出的文件。有关详细信息，请参阅第 7 章中的 Excel 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

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

```

excelexportnode 属性	数据类型	属性说明
full_filename	string	
excel_file_type	Excel2003 Excel2007	
export_mode	Create Append	
inc_field_names	flag	指定字段名是否可以包含在工作表的第一行中。
start_cell	string	指定导出的开始单元格。
worksheet_name	string	要写入的工作表的名称。
launch_application	flag	指定是否应在结果文件上调用 Excel。注意，必须在辅助应用程序对话框（“工具”菜单，辅助应用程序）中指定启动 Excel 的路径。
generate_import	flag	指定是否应生成将读取已导出数据文件的 Excel 导入节点。

outputfilenode 属性



平面文件导出节点将数据输出到已分隔的文本文件。这对导出可由其他分析或电子表格软件读取的数据非常有用。有关详细信息，请参阅第 7 章中的平面文件导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

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
```

outputfilenode 属性	数据类型	属性说明
full_filename	string	输出文件的名称。
write_mode	Overwrite Append	
inc_field_names	flag	
use_newline_after_records	flag	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	flag	
generate_import	flag	
encoding	StreamDefault SystemDefault "UTF-8"	

sasexportnode 属性



SAS 导出节点可以 SAS 格式输出数据，以便读入 SAS 或与 SAS 兼容的软件包中。有以下三种 SAS 文件格式：SAS for Windows/OS2、SAS for UNIX 或 SAS Version 7/8。有关详细信息，请参阅第 7 章中的 SAS 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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
```

sasexportnode 属性	数据类型	属性说明
format	Windows UNIX SAS7 SAS8	可变属性标签字段。
full_filename	string	
export_names	NamesAndLabels NamesAsLabels	用于将字段名从 IBM® SPSS® Modeler 的导出中映射到 IBM® SPSS® Statistics 或 SAS 的变量名中。
generate_import	flag	

statisticsexportnode 属性



Statistics 导出节点以 IBM® SPSS® Statistics.sav 格式输出数据。.sav 文件可由 SPSS Statistics Base 和其他产品读取。这种格式也用于 IBM® SPSS® Modeler 中的某些缓存文件。有关详细信息，请参阅第 8 章中的 Statistics 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

有关此节点属性的信息，请参阅 [statisticsexportnode 属性](#) 第 280 页码。

xmlexportnode 属性



XML 导出节点将数据以 XML 格式输出到文件。还可选择创建 XML 源节点，以将导出的数据读回到流中。有关详细信息，请参阅第 7 章中的 XML 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create xmlexportnode
set :xmlexportnode.full_filename = "c:\export\data.xml"
set :xmlexportnode.map = [{"/catalog/book/genre" genre} {"catalog/book/title" title}]
```

xmlexportnode 属性	数据类型	属性说明
full_filename	string	(必需) XML 导出文件的完整路径和文件名。
use_xml_schema	flag	指定是否使用 XML 架构 (XSD 或 DTD 文件) 控制导出数据的结构。
full_schema_filename	string	要使用的 XSD 或 DTD 文件的完整路径和文件名。如果 use_xml_schema 设为 true, 则为必需。
generate_import	flag	生成将已导出数据文件读回到流中的 XML 源节点。
records	string	表示记录边界的 XPath 表达式。
map	string	将字段名映射到 XML 结构。 示例: set :xmlexportnode.map = [{"/top/node1" field1} {"top/node2" field2}] 这将流字段 field1 映射到 XML 元素 /top/node1, 依此类推。

IBM SPSS Statistics 节点属性

statisticsimportnode 属性



Statistics 文件节点从 IBM® SPSS® Statistics 使用的 .sav 文件格式以及保存在 IBM® SPSS® Modeler 中的高速缓存文件（其也使用相同格式）读取数据。有关详细信息，请参阅第 8 章中的 Statistics 文件节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create statisticsimportnode
set :statisticsimportnode.full_filename = "C:/data/drug1n.sav"
set :statisticsimportnode.import_names = true
set :statisticsimportnode.import_data = true
```

statisticsimportnode 属性	数据类型	属性说明
full_filename	字符串	完整文件名（包括路径）。
import_names	NamesAndLabels LabelsAsNames	处理变量名和标签的方法。
import_data	DataAndLabels LabelsAsData	处理值和标签的方法。
use_field_format_for_storage	布尔逻辑	指定导入时是否使用 SPSS Statistics 字段格式信息。

statisticstransformnode 属性



Statistics 转换节点针对 IBM® SPSS® Modeler 中的数据源运行所选的 IBM® SPSS® Statistics 语法命令。此节点需要 SPSS Statistics 的许可副本。有关详细信息，请参阅第 8 章中的 Statistics 转换节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create statisticstransformnode
set :statisticstransformnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticstransformnode.new_name.NewVar = "Mixed Drugs"
set :statisticstransformnode.check_before_saving = true
```

statisticstransformnode 属性	数据类型	属性说明
syntax	字符串	
check_before_saving	标志	保存输入项之前验证已输入的语法。如果语法无效，则会显示一条错误消息。

statisticstransformnode 属性	数据类型	属性说明
default_include	标志	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。
include	标志	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。
new_name	字符串	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。

statisticsmodelnode 属性



Statistics 模型节点使您能够通过运行生成 PMML 的 IBM® SPSS® Statistics 过程分析和处理数据。此节点需要 SPSS Statistics 的许可副本。有关详细信息, 请参阅第 8 章中的 Statistics 模型节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
create statisticsmodelnode
set :statisticsmodelnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticsmodelnode.new_name.NewVar = "Mixed Drugs"
```

statisticsmodelnode 属性	数据类型	属性说明
syntax	字符串	
default_include	标志	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。
include	标志	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。
new_name	字符串	有关详细信息, 请参阅第 140 页码第 14 章中的 filternode 属性。

statisticsoutputnode 属性



Statistics 输出节点可调用 IBM® SPSS® Statistics 过程以分析您的 IBM® SPSS® Modeler 数据。可以访问许多不同的 SPSS Statistics 分析过程。此节点需要 SPSS Statistics 的许可副本。有关详细信息, 请参阅第 8 章中的 Statistics 输出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```
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 = "Cases by Age, Sex and Medical History"
set :statisticsoutputnode.file_type = HTML

```

statisticsoutputnode 属性	数据类型	属性说明
mode	Dialog Syntax	选择“SPSS Statistics 对话框”选项或语法编辑器
syntax	字符串	
use_output_name	标志	
output_name	字符串	
output_mode	Screen File	
full_filename	字符串	
file_type	HTML SPV SPW	

statisticsexportnode 属性



Statistics 导出节点以 IBM® SPSS® Statistics.sav 格式输出数据。 .sav 文件可由 SPSS Statistics Base 和其他产品读取。这种格式也用于 IBM® SPSS® Modeler 中的某些缓存文件。有关详细信息，请参阅第 8 章中的 Statistics 导出节点中的 IBM SPSS Modeler 15 源、过程和输出节点。

示例

```

create statisticsexportnode
set :statisticsexportnode.full_filename = "c:/output/SPSS_Statistics_out.sav"
set :statisticsexportnode.field_names = Names
set :statisticsexportnode.launch_application = True
set :statisticsexportnode.generate_import = True

```

statisticsexportnode 属性	数据类型	属性说明
full_filename	字符串	
launch_application	标志	
export_names	NamesAndLabels NamesAsLabels	用于将字段名从 SPSS Modeler 的导出中映射到 SPSS Statistics 或 SAS 的变量名中。
generate_import	标志	

超节点属性

下表中说明了超节点的特定属性。注意公共节点属性也可应用于超节点。

表 22-1
source_supernode

属性名称	属性类型/值列表	属性说明
parameters	任意	使用此属性可创建和访问在超节点的参数表中指定的参数。有关详细信息，请参阅下文。

表 22-2
process_supernode

属性名称	属性类型/值列表	属性说明
parameters	任意	使用此属性可创建和访问在超节点的参数表中指定的参数。有关详细信息，请参阅下文。

表 22-3
terminal_supernode

属性名称	属性类型/值列表	属性说明
parameters	任意	使用此属性可创建和访问在超节点的参数表中指定的参数。有关详细信息，请参阅下文。
execute_method	Script Normal	
script	字符串	

超节点参数

可使用通用格式在脚本中创建或设置超节点参数：

```
set mySuperNode.parameters.minvalue = 30
```

此外，除了指定（或不指定）超节点的名称外，还可以指定其类型：

```
set :process_supernode.parameters.minvalue = 30
```

```
set mySuperNode:process_supernode.parameters.minvalue = 30
```

还可以使用 CLEM 表达式设置参数值：

```
set :process_supernode.parameters.minvalue = "<expression>"
```

设置已封装节点的属性

通过创建与要设置的节点和属性的字面名称相匹配的超节点参数，可以为封装在超节点中的特定节点设置属性。例如，假设有一个源超节点，其中封装有变量文件节点以读取数据。可以按以下格式传递要读取的文件名（使用 `full_filename` 属性指定）：

```
set :source_supernode.parameters.':variablefilenode.full_filename' = "c:/mydata.txt"
```

以上创建了名为 `:variablefilenode.full_filename` 的超节点参数，且值为 `c:/mydata.txt`。假定在超节点中存在指定类型的节点，则会相应地设置此节点的已命名属性的值。注意，此操作是在流脚本中完成的一即，包括超节点的流的脚本—而不是在超节点脚本中完成的。请确保使用单引号来指定参数名。

此方法可用于任何已封装的节点，只要可生成有效的节点和属性引用。例如，要设置已封装样本节点的 `rand_pct` 属性，可使用下列任何格式：

```
set mySuperNode.parameters.':samplerand_pct' = 50
```

或

```
set mySuperNode.parameters.'Sample.rand_pct'= 50
```

或

```
set mySuperNode.parameters.'Sample:samplerand_pct'= 50
```

上述第一个引用假定流中仅有一个样本节点；第二个引用假定仅有一个名为“Sample”的节点但忽略其类型。第三个引用最明确，其中不仅指定了节点的名称还指定了节点的类型。

有关详细信息，请参阅第 9 章中的超节点参数中的 IBM SPSS Modeler 15 源、过程和输出节点。

超节点脚本的局限性。超节点无法处理其他流，也无法更改当前流。因此，在超节点脚本中无法使用应用于流的命令，例如 `open stream`、`get stream`、`execute_script` 等。

注意事项

这些信息开发用于在全球提供的产品和服务。

IBM 可能在其他国家/地区中不提供在本文档中讨论的产品、服务或功能。请咨询您当地的 IBM 代表以了解有关您所在地区当前可用产品和服务的信息。任何对 IBM 产品、程序或服务的引用，并不意味着仅可使用这些 IBM 产品、程序或服务。作为替代，可以使用任何功能相当的产品、程序或服务，前提是不侵犯任何 IBM 知识产权。不过，用户应自行负责评估和验证任何非 IBM 产品、程序或服务的工作情况。

在本文档中介绍的主题可能涉及 IBM 的专利或申请中的专利。您获得该文档并不意味着授予您任何这些专利许可。您可以将书面许可查询函件发送至：

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U. S. A.

要查询双字节字符集 (DBCS) 相关许可证信息，请联系所在国家/地区中的 IBM 知识产权部门，或者以书面形式将查询函件发送至：

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

以下段落不适用于英国或此类条款与当地法律不符的其他国家/地区： INTERNATIONAL BUSINESS MACHINES 公司“按原样”提供本出版物，不保证任何明示或暗示，包括但不限于对非侵权性、适销性或对特定用途适用性的暗示担保。有些州/省不允许针对某些交易的明示或暗示免责条款，因此本声明可能不适用于您。

本信息可能含有技术误差或排版错误。此处的信息会定期进行更改；这些更改会体现在本出版物的新版本中。IBM 可能随时对本出版物中所述的产品和/或程序进行改进和/或更改，恕不另行通知。

在本信息材料中对任何非 IBM 网站的引用仅为了方便用户，并不以任何方式表明对这些网站的认可。这些网站上的材料并非本 IBM 产品材料的一部分，您对这些网站的使用需自担风险。

IBM 可以自认为适当并且不会对您构成任何约束的任何方式使用或分发您提供的任何信息。

如果本程序的受许可方试图了解有关程序的信息以启用：(i) 在独立创建的程序和其他程序（包括本程序）之间交换信息；(ii) 相互使用交换的信息，则应联系：

IBM Software Group, Attention:Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

此类信息的提供应遵照相关条款和条件，其中包括在某些情况下支付适当费用。

在本文档中介绍的受许可保护程序，及其所有受许可保护材料由 IBM 在双方签署的“IBM 客户协议”、“IBM 国际程序许可证协议”或任何其他等同协议下提供。

此处所含的性能数据均在受控环境下决定。因此，在其他操作环境中获得的结果可能差异较大。有些测量可能在开发级的系统中进行，不保证这些测量结果与常用系统上的测量结果相同。此外，有些测量结果可能通过推断来估计得出。实际结果可能有所差异。此文档的用户应针对其具体环境验证适用的数据。

非 IBM 产品的相关信息来自这些产品的供应商，及其发布的公告或其他公开来源。IBM 并未测试过这些产品，无法确认有关非 IBM 产品的性能准确性、兼容性或任何其他声明。有关非 IBM 产品功能的问题应由这些产品的供应商负责。

有关 IBM 未来方向或意向的所有声明均可能未经通知即变更或撤销，并且仅代表目标和目的。

本信息包含日常业务运营中使用的数据和报告的示例。为了尽可能详尽地对其进行说明，示例中包含了人员的姓名、公司、品牌和产品的名称。所有这些名称均为虚构，与真实商业企业使用的名称和地址的任何雷同纯属巧合。

如果您正在查阅此信息的软拷贝，照片和彩色插图可能不会显示。

商标

IBM、IBM 徽标、ibm.com 和 SPSS 是 IBM Corporation 的商标，在全球许多司法辖区注册。有关最新的 IBM 商标列表，请访问网页 <http://www.ibm.com/legal/copytrade.shtml>。

Intel、Intel 徽标、Intel Inside、Intel Inside 徽标、Intel Centrino、Intel Centrino 徽标、Celeron、Intel Xeon、Intel SpeedStep、Itanium 和 Pentium 是 Intel Corporation 或其子公司在美国和其他国家或地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国、其他国家或地区或两者的注册商标。

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国、其他国家或地区或两者的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国、其他国家或地区或两者的商标。

其他产品和服务名称可能是 IBM 或其他公司的商标。



- abs 函数, 79
- aggreatenode 属性, 123
- allbutfirst 函数, 82
- allbutlast 函数, 82
- alphabefore 函数, 82
- analysisnode 属性, 257
- anomalydetectionnode 属性, 171
- anonymizenode 属性, 131
- appendnode 属性, 123
- applyanomalydetectionnode 属性, 224
- applyapriorinode 属性, 224
- applyautoclassifiernode 属性, 225
- applyautoclusternode 属性, 225
- applyautonumericnode 属性, 225
- applybayesnetnode 属性, 226
- applyc50node 属性, 226
- applycarmanode 属性, 226
- applycartnode 属性, 226
- applychaidnode 属性, 227
- applycoxregnode 属性, 227
- applydb2imclusternode 属性, 249
- applydb2imlognode 属性, 249
- applydb2imbnnode 属性, 249
- applydb2imregnode 属性, 249
- applydb2imtreenode 属性, 249
- applydecisionlistnode 属性, 228
- applydiscriminantnode 属性, 228
- applyfactornode 属性, 228
- applyfeatureselectionnode 属性, 228
- applygeneralizedlinearnode 属性, 229
- applykmeansnode 属性, 229
- applyknnnode 属性, 229
- applykohonennode 属性, 229
- applylinearnode 属性, 229
- applylogregnode 属性, 230
- appliesmlogisticnode 属性, 237
- appliesneuralnetworknode 属性, 237
- appliesregressionnode 属性, 236
- appliessequenceclusternode 属性, 237
- appliesmstimeseriesnode 属性, 237
- appliesstreenode 属性, 236
- appliesnetezabayesnode 属性, 256
- appliesnetezadectreenode 属性, 256
- appliesnetezadivclusternode 属性, 256
- appliesnetezakmeansnode 属性, 256
- appliesnetezaknnnode 属性, 256
- appliesnetezalinerregressionnode 属性, 256
- appliesnetezanaivebayesnode 属性, 256
- appliesnetezapcanode 属性, 256
- appliesnetezaregtreenode 属性, 256
- appliesneuralnetnode 属性, 230
- appliesneuralnetworknode 属性, 230
- applyoraabnnode 属性, 243
- applyoradecisiontreenode 属性, 243
- applyorakmeansnode 属性, 243
- applyoranbnnode 属性, 243
- applyoranmfnode 属性, 244
- applyoraoclusternode 属性, 243
- applyorasvmnode 属性, 243
- applyquestnode 属性, 231
- applyregressionnode 属性, 231
- applyselflearningnode 属性, 231
- applysequencenode 属性, 231
- appliesvmnode 属性, 232
- appliesmstimeseriesnode 属性, 232
- appliesmstepnode 属性, 232
- Apriori 模型
- 节点脚本属性, 172, 224
- apriorinode 属性, 172
- arccos 函数, 80
- arccosh 函数, 80
- arcsin 函数, 80
- arcsinh 函数, 80
- arctan 函数, 80
- arctan2 函数, 80
- arctanh 函数, 80
- autoclassifiernode 属性, 174
- autoclusternode 属性, 176
- autodataprepnode 属性, 132
- autonumericnode 属性, 178
- balancenode 属性, 124
- bayesnet 属性, 179
- binningnode 属性, 135
- @BLANK 函数, 76, 94
- C&R 树模型
- 节点脚本属性, 183, 226
- C5.0 模型
- 节点脚本属性, 180, 226
- c50node 属性, 180
- CARMA 模型
- 节点脚本属性, 181, 226
- carmanode 属性, 181
- cartnode 属性, 183
- cdf_chisq 函数, 81
- cdf_f 函数, 81
- cdf_normal 函数, 81
- cdf_t 函数, 81
- CHAID 模型
- 节点脚本属性, 185, 227
- chaidnode 属性, 185
- clear generated palette 命令, 41, 56
- clear stream 命令, 45
- CLEM
- 数据类型, 70 - 71
- 脚本编写, 8, 18
- 表达式, 69
- 语言, 69
- CLEM 函数
- sequence, 90 - 91
- string, 82
- 三角法, 80
- 信息, 76

索引

- 全局, 94
- 数值, 79
- 日期时间, 87
- 概率, 81
- 比较, 77
- 特殊函数, 95
- 空值与 Null 值, 94
- 逐位, 81
- 逻辑, 79
- 随机, 82
- CLEM 表达式
 - 参数, 23
 - 脚本编写, 30
- CLEM函数
 - 可用列表, 74
 - 转换, 76
- CLEM表达式
 - 查找并替换文本, 15
 - 脚本编写, 25
- close FILE 命令, 49
- close STREAM 命令, 44
- cognosimportnode 属性, 108
- collectionnode 属性, 158
- column_count 属性, 49
- connect NODE 命令, 34
- cos 函数, 80
- cosh 函数, 80
- count_equal 函数, 77
- count_greater_than 函数, 77
- count_less_than 函数, 77
- count_non_nulls 函数, 77
- count_not_equal 函数, 77
- count_nulls 函数, 77
- count_substring 函数, 82
- Cox 回归模型
 - 节点脚本属性, 187, 227
- coxregnode 属性, 187
- create NODE 命令, 33
- create stream 命令, 44
- dataauditnode 属性, 258
- databaseexportnode 属性, 270
- databasenode 属性, 110
- datacollectionexportnode 属性, 274
- datacollectionimportnode 属性, 111
- date 函数, 71-72
 - date_before, 77, 87
 - date_days_difference, 87
 - date_in_days, 87
 - date_in_months, 87
 - date_in_weeks, 87
 - date_in_years, 87
 - date_months_difference, 87
 - date_weeks_difference, 87
 - date_years_difference, 87
 - @TODAY 函数, 87
- date_before 函数, 77
- datetime_date 函数, 76
- db2imassocnode 属性, 245
- db2imclusternode 属性, 247
- db2imlognode 属性, 248
- db2immbnode 属性, 248
- db2imregnode 属性, 246
- db2imsequencenode 属性, 245
- db2imtimeseriesnode 属性, 248
- db2imtreenode 属性, 244
- decisionlist 属性, 189
- delete model 命令, 41, 51
- delete NODE 命令, 35
- derivnode 属性, 137
- DIFF 函数, 91
 - @DIFF 函数, 90-91
- directedwebnode 属性, 168
- disable NODE 命令, 35
- disconnect NODE 命令, 35
- discriminant 模型
 - 节点脚本属性, 190, 228
- discriminantnode 属性, 190
- distinctnode 属性, 125
- distributionnode 属性, 159
- div 函数, 79
- duplicate NODE 命令, 35
- enable NODE 命令, 35
- endstring 函数, 82
- ensemblenode 属性, 138
- Enterprise View 节点
 - 属性, 114
- evaluationnode 属性, 159
- evimportnode 属性, 114
- Excel 导出节点
 - 属性, 274
- Excel 源节点
 - 属性, 113
- excelexportnode 属性, 274
- excelimportnode 属性, 113
- execute NODE 命令, 36
- execute_all 命令, 28
- execute_project 命令, 48
- execute_script 命令, 28
- exit 命令, 24, 28
- export model 命令, 42
- export NODE 命令, 36
- export output 命令, 51
- f 分布
 - 概率函数, 81
- factornode 属性, 191
- featureselectionnode 属性, 12, 193
- @FIELD 函数, 95
- @FIELDS_BETWEEN 函数, 95
- @FIELDS_MATCHING 函数, 95
- fillernode 属性, 139
- filternode 属性, 140
- first_index 函数, 77
- first_non_null 函数, 77
- first_non_null_index 函数, 77

- fixedfilenode 属性, 115
- flatfilenode 属性, 275
- flush NODE 命令, 37
- for 命令, 22, 24, 53, 58, 61
- for...endfor 命令, 29
- fracof 函数, 79
- genlinnode 属性, 194
- get node 命令, 37
- get output 命令, 51
- get stream 命令, 45
- get 命令, 20
- GLMM 模型
 - 节点脚本属性, 197
- GLMM 节点属性, 197
- graphboardnode 属性, 161
- hasendstring 函数, 82
- hasmidstring 函数, 82
- hasstartstring 函数, 82
- hassubstring 函数, 82
- histogramnode 属性, 163
- historynode 属性, 141
- HTML 格式
 - 导出模型, 42
 - 导出节点, 36
- HTML 输出
 - 使用脚本创建, 58, 61
- IBM Cognos BI 源节点
 - 属性, 108
- IBM DB2 模型
 - 节点脚本属性, 244
- IBM ISW Logistic 回归模型
 - 节点脚本属性, 248 - 249
- IBM ISW Naive Bayes 模型
 - 节点脚本属性, 248 - 249
- IBM ISW Time Series 模型
 - 节点脚本属性, 248
- IBM ISW 关联模型
 - 节点脚本属性, 245, 249
- IBM ISW 决策树模型
 - 节点脚本属性, 244, 249
- IBM ISW 回归模型
 - 节点脚本属性, 246, 249
- IBM ISW 序列模型
 - 节点脚本属性, 245, 249
- IBM ISW 聚类模型
 - 节点脚本属性, 247, 249
- IBM SPSS Collaboration and Deployment Services Repository
 - 命令行参数, 65
 - 脚本编写, 54
- IBM SPSS Data Collection 导出节点
 - 属性, 274
- IBM SPSS Data Collection 源节点
 - 属性, 111
- IBM SPSS Modeler, 1
 - 从命令行运行, 63
 - 文档, 3
- IBM SPSS Statistics 导出节点
 - 属性, 280
- IBM SPSS Statistics 源节点
 - 属性, 278
- IBM SPSS Statistics 转换节点
 - 属性, 278
- IBM SPSS Statistics模型
 - 节点脚本属性, 279
- IBM SPSS Statistics输出节点
 - 属性, 279
- if 命令, 24, 58
- if...then...else 命令, 30
- if、then、else 函数, 79
- INDEX 函数, 91
- @INDEX 函数, 90 - 91
- insert model 命令, 43
- integer_bitcount 函数, 81
- integer_leastbit 函数, 81
- integer_length 函数, 81
- intof 函数, 79
- is_date 函数, 76
- is_datetime 函数, 76
- is_integer 函数, 76
- is_number 函数, 76
- is_real 函数, 76
- is_string 函数, 76
- is_time 函数, 76
- is_timestamp 函数, 76
- isalphacode 函数, 82
- isendstring 函数, 82
- islowercode 函数, 82
- ismidstring 函数, 82
- isnumbercode 函数, 82
- isstartstring 函数, 82
- issubstring 函数, 82
- issubstring_count 函数, 82
- issubstring_lim 函数, 82
- isuppercode 函数, 82
- K-Means 模型
 - 节点脚本属性, 200
- K-均值模型
 - 节点脚本属性, 229
- kmeansnode 属性, 200
- KNN 模型
 - 节点脚本属性, 229
- knnnode 属性, 201
- kohonen 模型
 - 节点脚本属性, 202
- Kohonen 模型
 - 节点脚本属性, 229
- kohonnenode 属性, 202
- last_index 函数, 77
- LAST_NON_BLANK 函数, 91
- @LAST_NON_BLANK 函数, 90 - 91, 94
- last_non_null 函数, 77
- last_non_null_index 函数, 77
- length 函数, 82

索引

- load model 命令, 43
- load node 命令, 37
- load output 命令, 52
- load project 命令, 48
- load state 命令, 49
- load stream 命令, 45
- locchar 函数, 82
- locchar_back 函数, 82
- Logistic 回归模型。
 - 节点脚本属性, 205, 230
- logregnode 属性, 205
- lowertoupper 函数, 53, 82
- matches 函数, 82
- matrixnode 属性, 259
- max 函数, 77
- MAX 函数, 91
- @MAX 函数, 90 - 91
- max_index 函数, 77
- max_n 函数, 77
- MEAN 函数, 90 - 91
- @MEAN 函数, 90 - 91
- mean_n 函数, 79
- meansnode 属性, 261
- member 函数, 77
- mergenode 属性, 126
- Microsoft 模型
 - 节点脚本属性, 233, 236
- min 函数, 77
- MIN 函数, 91
- @MIN 函数, 90 - 91
- min_index 函数, 77
- min_n 函数, 77
- mod 函数, 79
- MS Logistic 回归
 - 节点脚本属性, 233, 237
- MS 决策树
 - 节点脚本属性, 233, 236
- MS 序列聚类
 - 节点脚本属性, 237
- MS 时间序列
 - 节点脚本属性, 237
- MS 神经网络
 - 节点脚本属性, 233, 237
- MS 线性回归
 - 节点脚本属性, 233, 236
- msassocnode 属性, 233
- msbayesnode 属性, 233
- msclusternode 属性, 233
- mslogisticnode 属性, 233
- msneuralnetworknode 属性, 233
- msregressionnode 属性, 233
- mssequenceclusternode 属性, 233
- mstimeseriesnode 属性, 233
- mstreenode 属性, 233
- @MULTI_RESPONSE_SET 函数, 95
- multiplotnode 属性, 164
- netezza GLM 节点属性, 255
- Netezza K-Means 模型
 - 节点脚本属性, 251, 256
- Netezza KNN 模型
 - 节点脚本属性, 252, 256
- Netezza Naive Bayes 模型
 - 节点脚本属性, 252, 256
- Netezza PCA 模型
 - 节点脚本属性, 253, 256
- Netezza 决策树模型
 - 节点脚本属性, 250, 256
- Netezza 分裂式聚类模型
 - 节点脚本属性, 252, 256
- Netezza 回归树模型
 - 节点脚本属性, 253, 256
- Netezza 广义线性模型
 - 节点脚本属性, 255
- Netezza 时间序列模型
 - 节点脚本属性, 254
- netezza 时间序列节点属性, 254
- Netezza 模型
 - 节点脚本属性, 250
- Netezza 线性回归模型
 - 节点脚本属性, 254, 256
- Netezza 贝叶斯网络模型
 - 节点脚本属性, 251, 256
- netezzabayesnode 属性, 251
- netezadectreenode 属性, 250
- netezadivclusternode 属性, 252
- netezzakmeansnode 属性, 251
- netezzaknnode 属性, 252
- netezzalineressionnode 属性, 254
- netezzanaivebayesnode 属性, 252
- netezzapcanode 属性, 253
- netezzaregtreenode 属性, 253
- neuralnetnode 属性, 209
- neuralnetworknode 属性, 211
- @NULL 函数, 76, 94
- numericpredictornode 属性, 178
- OFFSET 函数, 91
- @OFFSET 函数, 90 - 91
- oneof 函数, 82
- open FILE 命令, 50
- open stream 命令, 22, 46
- oraabnode 属性, 239
- oraainode 属性, 242
- oraapriorinode 属性, 242
- Oracle Adaptive Bayes 模型
 - 节点脚本属性, 239, 243
- Oracle AI 模型
 - 节点脚本属性, 242
- Oracle Apriori 模型
 - 节点脚本属性, 242, 244
- Oracle KMeans 模型
 - 节点脚本属性, 241, 243
- Oracle MDL models
 - 节点脚本属性, 242, 244

- Oracle Naive Bayes 模型
 - 节点脚本属性, 238, 243
- Oracle NMF 模型
 - 节点脚本属性, 241, 244
- Oracle O-Cluster
 - 节点脚本属性, 241, 243
- Oracle Support Vector Machines 模型
 - 节点脚本属性, 239, 243
- Oracle 决策树模型
 - 节点脚本属性, 240, 243
- Oracle 广义线性模型
 - 节点脚本属性, 240
- Oracle 模型
 - 节点脚本属性, 238
- oradecisiontreenode 属性, 240
- oraglmnode 属性, 240
- orakmeansnode 属性, 241
- oramdlnode 属性, 242
- oranbnode 属性, 238
- oranmfnode 属性, 241
- oraoclusternode 属性, 241
- orasvmnode 属性, 239
- outputfilenode 属性, 275
- parameters
 - 超节点, 281
- @PARTITION_FIELD 函数, 95
- partitionnode 属性, 142
- PCA 模型
 - 节点脚本属性, 191, 228
- pi 函数, 80
- plotnode 属性, 165
- PMML 格式
 - 导出模型, 42
 - 导出节点, 36
- position NODE 命令, 37
- @PREDICTED 函数, 95
- QUEST 模型
 - 节点脚本属性, 213, 231
- questnode 属性, 213
- random 函数, 82
- random0 函数, 82
- reclassifynode 属性, 143
- regressionnode 属性, 215
- rem 函数, 79
- rename NODE 命令, 22, 38
- reordernode 属性, 144
- replace 函数, 82
- replicate 函数, 82
- reportnode 属性, 262
- restructurenode 属性, 144
- retrieve model 命令, 43
- retrieve node 命令, 38
- retrieve output 命令, 52
- retrieve project 命令, 48
- retrieve stream 命令, 46
- retrieve 命令, 54
- RFM 分析节点
 - 属性, 145
- RFM 汇总节点
 - 属性, 127
- rfmaggregatenode 属性, 127
- rfmanalysisnode 属性, 145
- row_count 属性, 49
- samplenode 属性, 128
- SAS 导出节点
 - 属性, 276
- SAS 源节点
 - 属性, 117
- sasexportnode 属性, 276
- sasimportnode 属性, 117
- save model 命令, 44
- save node 命令, 38
- save output 命令, 52
- save project 命令, 48
- save STREAM 命令, 46
- save 命令, 20
- SDEV 函数, 91
- @SDEV 函数, 90 - 91
- sdev_n 函数, 79
- selectnode 属性, 130
- sequencenode 属性, 217
- set 命令, 19, 22 - 23, 30
- setglobalsnode 属性, 263
- settoflagnode 属性, 146
- sin 函数, 80
- SINCE 函数, 91
- @SINCE 函数, 90 - 91
- sinh 函数, 80
- skipchar 函数, 82
- skipchar back 函数, 82
- SLRM 模型
 - 节点脚本属性, 218, 231
- slrmnode 属性, 218
- sortnode 属性, 130
- SoundEx 函数, 86
- soundex_difference 函数, 86
- SPSS Modeler Server, 2
- SQL 格式
 - 导出节点, 36, 42
- startstring 函数, 82
- statisticsexportnode 属性, 280
- statisticsimportnode 属性, 12, 278
- statisticsmodelnode 属性, 279
- statisticsnode 属性, 264
- statisticsoutputnode 属性, 279
- statisticstransformnode 属性, 278
- store model 命令, 44
- store node 命令, 39
- store output 命令, 52
- store project 命令, 48
- store stream 命令, 47
- store 命令, 54
- stream.nodes 属性, 53

索引

- stripchar 函数, 82
- strmember 函数, 82
- subscrs 函数, 82
- substring 函数, 82
- substring_between 函数, 82
- SUM 函数, 91
- @SUM 函数, 90 - 91
- sum_n 函数, 79
- Support vector machine 模型
 - 节点脚本属性, 219
- Support Vector Machine 模型
 - 节点脚本属性, 232
- SVM 模型
 - 节点脚本属性, 219
- svmnode 属性, 219
- t 分布
 - 概率函数, 81
- tablnode 属性, 265
- tan 函数, 80
- tanh 函数, 80
- @TARGET 函数, 95
- testbit 函数, 81
- @TESTING_PARTITION 函数, 95
- THIS 函数, 91
- @THIS 函数, 90 - 91
- time_before 函数, 77
- timeintervalsnode 属性, 147
- timeplotnode 属性, 167
- timeseriesnode 属性, 220
- to_date 函数, 76, 87
- to_dateline 函数, 87
- to_datetime 函数, 76
- to_integer 函数, 76
- to_number 函数, 76
- to_real 函数, 76
- to_string 函数, 76
- to_time 函数, 76, 87
- to_timestamp 函数, 76, 87
- @TODAY 函数, 87
- @TRAINING_PARTITION 函数, 95
- transformnode 属性, 268
- transposenode 属性, 151
- trim 函数, 82
- trim_start 函数, 82
- trimend 函数, 82
- twostepnode 属性, 222
- typenode 属性, 12, 59, 152
- undef 函数, 94
- unicode_char 函数, 82
- unicode_value 函数, 82
- uppertolower 函数, 82
- userinputnode 属性, 118
- @VALIDATION_PARTITION 函数, 95
- value 命令, 49
- value_at 函数, 77
- var 命令, 19, 22, 33
- variablefilenode 属性, 119
- Web 节点
 - 属性, 168
- webnode 属性, 168
- with stream 命令, 22, 47
- write FILE 命令, 50
- writeln FILE 命令, 50, 58, 61
- XML 导出节点
 - 属性, 277
- XML 源节点
 - 属性, 121
- xmlexportnode 属性, 277
- xmlimportnode 属性, 121
- 三角函数, 80
- 不等运算符, 77
- 与运算符, 79
- 两步模型
 - 节点脚本属性, 222, 232
- 中断脚本, 14
- 主成分分析 (PCA) / 因子模型
 - 节点脚本属性, 191, 228
- 优先级, 73
- 会话参数, 23, 30
- 位元函数, 81
- 信息函数, 76
- 值属性, 60
- 全局函数, 94
- 决策列表模型
 - 节点脚本属性, 189, 228
- 函数, 71 - 72, 75 - 76, 90
 - @FIELD, 95
 - @GLOBAL_MAX, 94
 - @GLOBAL_MEAN, 94
 - @GLOBAL_MIN, 94
 - @GLOBAL_SDEV, 94
 - @GLOBAL_SUM, 94
 - @PARTITION, 95
 - @PREDICTED, 95
 - @TARGET, 95
- 分区节点
 - 属性, 142
- 分布函数, 81
- 分析节点
 - 属性, 257

- 分级节点
 - 属性, 135
- 列出参数
 - 在脚本中进行修改, 25
 - 列表, 69, 71
- 加密密码
 - 添加至脚本, 55
- 匿名化节点
 - 属性, 131
- 卡方分布
 - 概率函数, 81
- 历史节点
 - 属性, 141
- 参数, 13, 30, 98, 100, 102
 - IBM SPSS Collaboration and Deployment Services Repository 连接, 65
 - 会话, 23
 - 命令文件, 64
 - 服务器连接, 64
 - 流, 23
 - 系统, 66
 - 脚本编写, 18, 25
- 取整函数, 79
- 变换节点
 - 属性, 268
- 变量, 22, 30
 - 脚本编写, 18, 20
 - 节点引用, 19
- 变量文件节点
 - 属性, 119
- 合并节点
 - 属性, 126
- 名义字段
 - 值属性, 60
- 命令行
 - 参数, 67
 - 参数列表, 64 - 66
 - 启动 IBM SPSS Modeler, 63
 - 多个参数, 64
 - 脚本编写, 56
 - 运行 IBM SPSS Modeler, 63
- 商标, 284
- 固定文件节点
 - 属性, 115
- 图形板节点
 - 属性, 161
- 图形节点
 - 脚本属性, 157
- 在 CLEM 表达式中的反斜杠字符, 70
- 块
 - 节点脚本属性, 224
- 填充节点
 - 属性, 139
- 多重散点图节点
 - 属性, 164
- 多重设置命令, 98
- 大于运算符, 77
- 字段, 69, 71
 - 在脚本中关闭, 157
- 字段名
 - 更改大小写, 53
- 字段重排节点
 - 属性, 144
- 字符, 69 - 70
- 字符串, 69 - 70
 - 更改大小写, 53
 - 脚本编写, 19
- 字符串函数, 53, 82
- 安全
 - 加密密码, 55, 64
- 定向 Web 节点
 - 属性, 168
- 实数, 69 - 70
- 密码
 - 已编码, 64
 - 添加至脚本, 55
- 导出
 - PMML, 36, 42
 - SQL, 36, 42
 - 模型, 42
 - 节点, 36
- 导出节点
 - 节点脚本属性, 269
- 小于运算符, 77
- 局部变量, 22, 30
- 属性, 30
 - 工程, 105
 - 数据库建模节点, 233
 - 流, 102
 - 脚本编写, 98, 100, 171, 224, 269
 - 超节点, 281
 - 过滤节点, 98

索引

- 通用脚本编写, 100
- 工程
 - 属性, 105
- 差异节点
 - 属性, 125
- 已生成的模型
 - 脚本编写名称, 39, 41
- 常用对数函数, 79
- 幂(指数)函数, 79
- 平均值节点
 - 属性, 261
- 平方根函数, 79
- 平衡节点
 - 属性, 124
- 平面文件节点
 - 属性, 275
- 广义线性模型
 - 节点脚本属性, 194, 229
- 序列函数, 90 - 91
- 序列模型
 - 节点脚本属性, 217, 231
- 应用程序示例, 3
- 建模节点
 - 节点脚本属性, 171
- 异常检测模型
 - 节点脚本属性, 171, 224
- 当前对象
 - 在脚本中引用, 20
- 循环
 - 在脚本中使用, 53, 59, 61
- 或运算符, 79
- 执行脚本, 14
- 执行顺序
 - 用脚本更改, 53
- 报告
 - 使用脚本创建, 58, 61
- 报告节点, 58, 61
 - 属性, 262
- 指数函数, 79
- 排序节点
 - 属性, 130
- 插入符语法
 - 变量引用, 19, 22
- 收集节点
 - 属性, 158
- 散点图节点
 - 属性, 165
- 数值函数, 79
- 数字, 70
- 数据审核节点
 - 属性, 258
- 数据库导出节点
 - 属性, 270
- 数据库建模, 233
- 数据库节点
 - 属性, 110
- 整体节点
 - 属性, 138
- 整数, 69 - 70
- 文件对象
 - 脚本编写命令, 49
- 文字
 - 脚本编写, 18, 26
- 文字字符串
 - 嵌入脚本中, 26
- 文本字符串
 - 嵌入脚本中, 26
- 文本格式
 - 导出模型, 42
 - 导出节点, 36
- 文档, 3
- 日期
 - 操作, 89
 - 转换, 89
- 日期时间函数
 - datetime_date, 87
 - datetime_day, 87
 - datetime_day_name, 87
 - datetime_day_short_name, 87
 - datetime_hour, 87
 - datetime_in_seconds, 87
 - datetime_minute, 87
 - datetime_month, 87
 - datetime_month_name, 87
 - datetime_month_short_name, 87
 - datetime_now datetime_second, 87
 - datetime_time, 87
 - datetime_timestamp, 87
 - datetime_weekday, 87
 - datetime_year, 87
- 日期格式, 71 - 72
- 时间函数, 71 - 72
 - time_before, 77, 87
 - time_hours_difference, 87

- time_in_hours, 87
- time_in_mins, 87
- time_in_secs, 87
- time_mins_difference, 87
- time_secs_difference, 87
- 时间区间节点
 - 属性, 147
- 时间和日期函数, 71 - 72
- 时间字段
 - 转换, 89
- 时间序列模型
 - 节点脚本属性, 220, 232
- 时间散点图节点
 - 属性, 167
- 时间格式, 71 - 72

- 替换文本, 15
- 最近相邻元素模型
 - 节点脚本属性, 201

- 服务器
 - 命令行参数, 64

- 条形图节点
 - 属性, 159
- 查找文本, 15
- 标志
 - 命令行参数, 63
 - 组合多个标志, 264
- 标志字段
 - 值属性, 60
- 树生长指令
 - 嵌入脚本中, 26
- 样本节点
 - 属性, 128
- 概率函数, 81
- 模型
 - 导出, 42
 - 脚本编写, 42
 - 脚本编写名称, 39, 41
- 模型块
 - 脚本编写名称, 39, 41
 - 节点脚本属性, 224
- 模型对象
 - 脚本编写名称, 39, 41
 - 脚本编写命令, 39

- 正态分布
 - 概率函数, 81

- 比较函数, 77

- 求反函数, 79
- 法律注意事项, 283

- 注解
 - 在脚本中访问, 62
- 注释
 - 脚本编写, 26
- 派生节点
 - 属性, 137
- 流
 - 多重设置命令, 98
 - 属性, 102
 - 脚本编写, 8
- 流参数, 23, 30
- 流名称
 - 在脚本中访问, 61
- 流对象
 - 引用, 22
 - 打开, 21 - 22
 - 脚本编写命令, 44
- 流属性, 61
- 流执行顺序
 - 用脚本更改, 53
- 源节点
 - 属性, 106

- 特征选择模型
 - 应用, 12
 - 脚本编写, 12
 - 节点脚本属性, 193, 228
- 特殊函数, 95
- 特殊变量, 20

- 状态对象
 - 脚本编写命令, 49
- 独立脚本, 8, 10

- 生成的关键字, 56

- 用户输入节点
 - 属性, 118

- 直方图节点
 - 属性, 163
- 相等运算符, 77

- 矩阵节点
 - 属性, 259

- 示例
 - 应用程序指南, 3
 - 概述, 5
- 神经网络
 - 节点脚本属性, 211, 230
- 神经网络模型
 - 节点脚本属性, 209, 230

索引

- 空值处理
 - CLEM函数, 94
- 空格
 - 从字符串中删除, 82
- 空白
 - 从字符串中删除, 82
- 符号函数, 79
- 简介, 69
- 类型节点
 - 属性, 152
- 系统
 - 命令行参数, 66
- 约定, 75
- 线性回归模型
 - 节点脚本属性, 215, 231
- 线性属性, 204
- 线性模型
 - 节点脚本属性, 204, 229
- 结构化属性, 98
- 结果对象
 - 脚本编写命令, 49
- 统计量节点
 - 属性, 264
- 聚合节点
 - 属性, 123
- 脚本
 - 从文本文件导入, 8
 - 保存, 8
- 脚本编写
 - CLEM表达式, 25
 - syntax, 18
 - 与早期版本的兼容性, 56
 - 中断, 14
 - 从命令行, 56
 - 图形节点, 157
 - 在超节点中, 13
 - 当前对象, 20
 - 所用缩写, 99
 - 执行, 14
 - 执行脚本, 24
 - 查找并替换文本, 15
 - 概述, 8, 18
 - 注释, 26
 - 流, 8
 - 流执行顺序, 53
 - 特征选择模型, 12
 - 独立脚本, 8
 - 用户界面, 8, 10, 13
 - 示例, 58, 61
 - 节点, 19
 - 超节点脚本, 8
 - 输出节点, 257
 - 运算符, 25
 - 连接符, 26
 - 通用属性, 100
 - 错误检查, 56
- 自动分类器模型
 - 节点脚本属性, 225
- 自动分类器节点
 - 节点脚本属性, 174
- 自动数值模型
 - 节点脚本属性, 178, 225
- 自动数据准备
 - 属性, 132
- 自动聚类模型
 - 节点脚本属性, 225
- 自动聚类节点
 - 节点脚本属性, 176
- 自学响应模型。
 - 节点脚本属性, 218, 231
- 自然对数函数, 79
- 节点
 - 在脚本中循环, 53
- 节点 ID
 - 在脚本中引用, 19
- 节点对象
 - 脚本编写, 19
 - 脚本编写命令, 33
- 节点属性
 - 在脚本中访问, 62
- 节点脚本属性, 233
 - 导出节点, 269
 - 建模节点, 171
 - 模型块, 224
- 表节点
 - 属性, 265
- 表达式, 69
- 表达式构建器
 - 查找并替换文本, 15
- 设为标志节点
 - 属性, 146
- 设置全局节点
 - 属性, 263
- 评估节点
 - 属性, 159
- 贝叶斯网络模型
 - 节点脚本属性, 179, 226

- 超节点, 98
 - parameters, 281
 - 参数, 23, 30
 - 属性, 281
 - 脚本, 8, 13 - 14
 - 脚本编写, 281
 - 设置属性, 281

- 转换函数, 76
- 转置节点
 - 属性, 151
- 输出对象
 - 脚本编写名称, 50
 - 脚本编写命令, 50
- 输出节点
 - 脚本属性, 257

- 过滤节点
 - 属性, 140
- 运算符
 - 脚本编写, 25
 - 连接字符串, 61, 76
- 运算符优先级, 73
- 连接字符串, 76
- 连接符
 - 脚本编写, 26
- 连续字段
 - 值属性, 60
- 追加节点
 - 属性, 123
- 选择节点
 - 属性, 130
- 通道参数, 13, 30, 98, 100
- 逻辑函数, 79

- 重排节点
 - 属性, 144
- 重新分类节点
 - 属性, 143
- 重新结构化节点
 - 属性, 144

- 错误检查
 - 脚本编写, 56

- 非运算符, 79