

IBM SPSS Modeler 15 CLEF 开
发人员指南



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 概述	1
CLEF 简介	1
系统体系结构	1
客户端组件	1
服务器端组件	2
CLEF 的功能	3
规范文件	3
节点	4
Data Model	4
输入和输出文件	4
应用程序编程接口 (API)	5
文件结构	5
客户端组件	5
服务器端组件	7
2 节点	9
节点概述	9
数据读取器节点	10
数据变换器节点	11
模型构建器节点	11
文档构建器节点	12
模型填充器节点	13
数据记录器节点	13
菜单、工具栏和选项板	13
菜单和子菜单	13
工具栏	14
选项板和子选项板	15
设计节点图标	16
边框	17
背景	18
图形要求	18
创建自定义图像	18
将图像文件添加至节点规范	19
设计对话框	20
关于节点对话框	20
对话框的设计原则	21
对话框组件	21
设计输出窗口	26

3 CLEF 示例 29

关于示例	29
激活示例	29
数据读取器节点 (Apache Log Reader)	30
数据转换器节点 (URL Parser)	31
文档构建器节点 (Web Status Report)	31
模型构建器节点 (Interaction)	32
检查规范文件	32
检查源代码	33
删除示例	33

4 规范文件 34

规范文件概述	34
规范文件的示例	35
XML 声明	36
扩展元素	36
扩展详细信息部分	37
资源部分	38
数据包	38
Jar 文件	39
共享库	39
帮助信息	40
通用对象部分	40
属性类型	41
属性集	42
容器类型	43
Actions	44
Catalogs	46
用户界面 (选项板) 部分	48
对象定义部分	53
对象标识符	54
模型构建器	57
文档构建器	57
模型提供者	57
Properties	58
Containers	59
用户界面	61
执行	61

输出数据模型	66
Constructors	67
通用特征	67
值类型	67
求值的字符串	71
操作	72
字段和字段元数据	77
字段集	77
角色	79
逻辑运算符	79
条件	80
在脚本中使用 CLEF 节点	84
保持向后兼容性	85

5 构建模型和文档 86

模型和文档构建简介	86
模型	86
文档	86
Constructors	86
构建模型	87
模型构建器	87
模型输出	95
构建交互模型	96
自动建模	101
应用模型	107
构建文档	107
文档构建器	108
文档输出	108
使用构造函数	109
创建模型输出	110
创建文档输出	111
创建交互模型构建器	111
创建模型填充器	112

6 构建用户界面 113

关于用户界面	113
用户界面部分	116
Icons	118

Controls	119
菜单	120
菜单项	121
工具栏项	123
示例：添加至主窗口	123
Tabs	124
访问键和键盘快捷键	126
面板规范	127
文本浏览器面板	127
扩展对象面板	129
属性面板	131
模型查看器面板	133
属性控件规范	134
UI 组件控件	134
属性面板控件	138
控制器	141
属性控件布局	163
标准控件布局	163
自定义控件布局	165
自定义输出窗口	178

7 添加帮助系统 180

帮助系统的类型	180
HTML 帮助	180
JavaHelp	181
实施帮助系统	182
定义帮助系统的位置和类型	182
指定要显示的特定帮助主题	183

8 本地化和辅助功能 185

简介	185
本地化	185
属性文件	186
帮助文件	190
测试本地化后的 CLEF 节点	191
辅助功能	191

9 程序设计 193

关于 CLEF 节点的程序设计	193
CLEF API 文档	193
客户端 API	193
客户端 API 类.	193
使用客户端 API	194
Predictive Server API (PSAPI)	195
服务器端 API	195
体系结构.	195
服务函数.	195
回调函数.	197
处理流程.	199
服务器端 API 功能.	201
错误处理.	214
XML 解析 API	215
使用服务器端 API	215
服务器端编程准则	215

10 测试和分发 219

测试 CLEF 扩展.	219
测试 CLEF 扩展	219
调试 CLEF 扩展	219
分发 CLEF 扩展.	223
安装 CLEF 扩展.	223
卸载 CLEF 扩展.	223

附录

A CLEF XML 架构 224

CLEF 元素参考	224
Action 元素	224
ActionButton 元素	225
Actions 元素	225
AddField 元素.	226
And 元素.	230

Attribute 元素	230
BinaryFormat 元素	231
Catalog 元素	232
Catalogs 元素.	232
ChangeField 元素	233
CheckBoxControl 元素.	236
CheckBoxGroupControl 元素.	237
ClientDirectoryChooserControl 元素.	239
ClientFileChooserControl 元素	240
ComboBoxControl 元素.	241
Command 元素	242
CommonObjects 元素	243
Condition 元素	244
Constraint 元素.	246
Constructors 元素	247
Container 元素	247
ContainerFile 元素	248
ContainerTypes 元素	248
Controls 元素.	248
CreateDocument 元素	249
CreateDocumentOutput 元素.	250
CreateInteractiveDocumentBuilder 元素.	250
CreateInteractiveModelBuilder 元素	251
CreateModel 元素	252
CreateModelApplier 元素	253
CreateModelOutput 元素.	254
DatabaseConnectionValue 元素	255
DataFile 元素.	256
DataFormat 元素.	256
DataModel 元素	256
DBConnectionChooserControl 元素	263
DBTableChooserControl 元素	264
DefaultValue 元素	265
DelimitedDataFormat 元素	267
DisplayLabel 元素	268
DocumentBuilder 元素.	268
DocumentOutput 元素	269
DocumentType 元素	270
Enabled 元素	271
Enumeration 元素	272
ErrorDetail 元素	273
Executable 元素.	275
Execution 元素	275
Extension 元素	275
ExtensionDetails 元素	276

ExtensionObjectPanel 元素	277
Field 元素	278
FieldFormats 元素	281
FieldGroup 元素	282
FieldGroups 元素	283
FileFormatType 元素	285
FileFormatTypes 元素	286
ForEach 元素	286
Icon 元素	287
Icons 元素	287
InputFiles 元素	288
InteractiveDocumentBuilder 元素	288
InteractiveModelBuilder 元素	289
Layout 元素	291
License 元素	293
ListValue 元素	293
MapValue 元素	294
Menu 元素	298
MenuItem 元素	300
MissingValues 元素	301
ModelBuilder 元素	304
ModelOutput 元素	313
ModelProvider 元素	315
ModelType 元素	315
ModelViewerPanel 元素	316
Module 元素	317
MultiFieldChooserControl 元素	317
MultiltemChooserControl 元素	319
Node 元素	320
Not 元素	322
NumberFormat 元素	323
NumericInfo 元素	324
Option 元素	324
OptionCode 元素	325
Or 元素	325
OutputDataModel 元素	326
OutputFiles 元素	327
Palette 元素	327
Parameters 元素	328
PasswordBoxControl 元素	330
Properties 元素	331
PropertiesPanel 元素	331
PropertiesSubPanel 元素	333
Property 元素	334
PropertyControl 元素	336

PropertyGroup 元素	337
PropertySets 元素	337
PropertyType 元素	338
PropertyTypes 元素	339
RadioButtonGroupControl 元素	339
Range 元素	341
Range 元素	341
RemoveField 元素	342
Resources 元素	342
Run 元素	345
SelectorPanel 元素	345
ServerDirectoryChooserControl 元素	346
ServerFileChooserControl 元素	348
SetContainer 元素	349
SetProperty 元素	350
SingleFieldChooserControl 元素	350
SingleFieldValueChooserControl 元素	352
SingleItemChooserControl 元素	353
SpinnerControl 元素	354
SPSSDataFormat 元素	355
StaticText 元素	356
StatusCode 元素	356
StatusCodes 元素	357
StatusDetail 元素	357
Structure 元素	360
StructuredValue 元素	360
SystemControls 元素	362
Tab 元素	363
TabbedPanel 元素	364
TableControl 元素	364
Tabs 元素	366
TextAreaControl 元素	366
TextBoxControl 元素	368
TextBrowserPanel 元素	369
ToolBarItem 元素	370
UserInterface 元素	371
UTF8Format 元素	371
Value 元素	371
Values 元素	372
Values 元素	373
Visible 元素	374
扩展 类型	374

B 注意事项	376
索引	378

概述

CLEF 简介

组件级扩展框架（CLEF）是一种允许向 IBM® SPSS® Modeler 的标准功能添加用户提供的扩展的机制。扩展通常包含可添加到 SPSS Modeler 中的共享库（例如，数据处理例程或建模算法），并且可通过某个菜单上的新条目或节点选项板上的新节点访问该库。

要执行该操作，SPSS Modeler 需要有关该自定义程序的详细信息，例如其名称、应传递给该程序的命令参数以及 SPSS Modeler 如何向程序显示选项和如何向用户显示结果等。要提供此信息，您应当提供 XML 格式的文件，即**规范文件**。SPSS Modeler 会将该文件中的信息转换为新菜单条目或节点定义。

使用 CLEF 的好处包括：

- 提供简单易用、异常灵活且稳定的环境，可供工程师、顾问和最终用户将新功能集成到 SPSS Modeler 中。
- 确保扩展模块的外观和功能与本地 SPSS Modeler 模块的相同。
- 使扩展节点具有与本地 SPSS Modeler 节点尽可能接近的执行速度和效率。

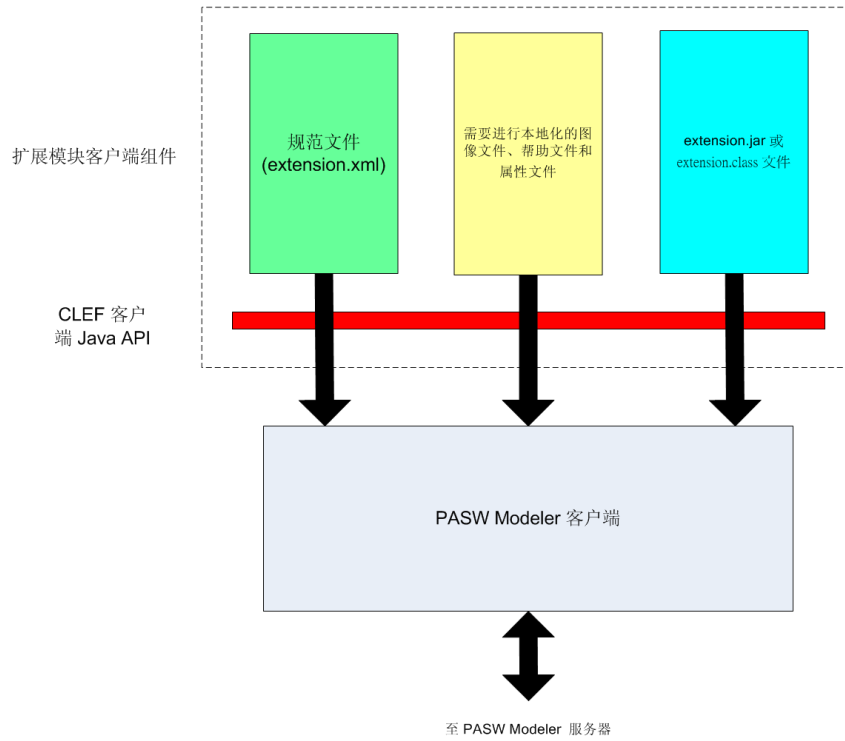
系统体系结构

与 IBM® SPSS® Modeler 自身相似，CLEF 使用两层的客户端/服务器体系结构，在该结构中，这两层可位于同一计算机中，也可位于两台不同的计算机上。

客户端组件

此处显示了客户端层的组件。

图片 1-1
客户端组件

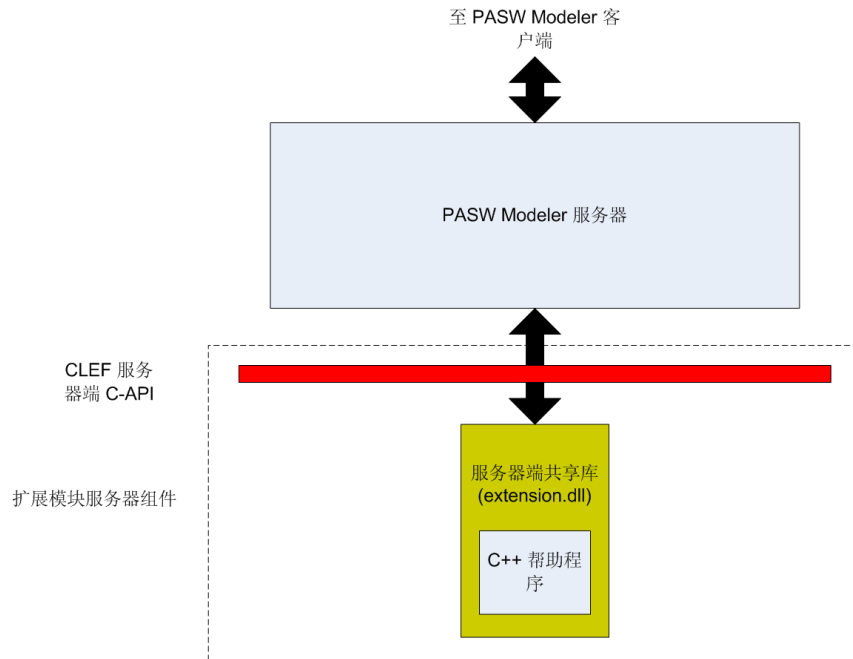


- **规范文件。** 列出了扩展定义的属性、格式、数据模型更改、控件和其他特征。
- **图像文件。** 包含用于识别扩展中的节点的图像。
- **帮助文件。** 用于显示有关扩展的帮助信息。
- **属性文件。** 包含的文本字符串包括扩展在屏幕上显示的名称、标签和消息。
- **Java .jar 或 .class 文件。** 包含了扩展使用的所有 Java 资源。
- **Java 应用程序编程接口 (API)。** 可供需要未由规范文件直接提供的其他控件、用户界面组件或交互的扩展使用。

服务器端组件

此处显示了服务器层的组件。

图片 1-2
服务器端组件



- **共享库的基于 C 的 API。**包含设置和获取执行设置、这些设置的持续时间、执行反馈、作业控制（例如，中断执行）、SQL 生成和返回的对象等方面。
- **服务器端共享库。**一个支持节点执行的动态链接库（DLL）。C++ 工具是一些基于 C 的 API 的包装器，这些基于 C 的 API 作为源代码提供，可以很容易地编译为 C++ CLEF 模块。

CLEF 的功能

下面的章节介绍了 CLEF 的许多主要功能：

- 规范文件
- 节点
- 数据模型
- 输入和输出文件
- 应用程序编程接口（API）

规范文件

CLEF 规范文件是一个 XML 文件，它包含说明新扩展行为的结构规范。规范文件说明的内容包括：

- 扩展所需的共享资源（例如，本地化后的文本数据包和服务器端的共享库）。

- 通用定义，例如文件类型或属性类型。
- 最终用户可以使用的对象，例如节点和输出模型。

启动 IBM® SPSS® Modeler 后，将从规范文件所在的位置导入该文件，这样可以立即使用该文件中定义的功能。

有关更多信息，请参阅第 4 章和规范文件第 34 页码。

节点

将扩展添加到实现新节点的 IBM® SPSS® Modeler 中时，需要先确定要创建的节点类型（例如，节点是用于生成模型还是只用于转换数据）。有关详细信息，请参阅第 9 页码第 2 章中的节点概述。

创建规范文件和所有必需的 Java 类和共享库后，将文件复制到 SPSS Modeler 可读取这些文件的特定位置。下次启动 SPSS Modeler 时，新节点将添加到相应的选项板中，并随时可用。

Data Model

数据模型表示流过 IBM® SPSS® Modeler 流的数据的结构。该模型描述流中各个点的数据，它对应于“类型”节点中的信息。它列出了流中特定点处现有字段的名称，并说明了这些字段的类型。

将任何节点添加到 SPSS Modeler 时，请考虑传输到此节点的数据模型对节点行为的影响。例如，一个衍生节点先获取输入数据模型，再为模型添加新字段，随后生成输出数据模型并将其传递到 SPSS Modeler 流中的下一个节点中。与此相比，图形节点先获取输入数据模型，但不生成任何输出数据模型，因为该数据不会被传递到任何后续节点。SPSS Modeler 必须知道要对数据模型执行的操作，这样后续节点可以显示有关哪些字段可用的正确信息。规范文件中的数据模型信息是 SPSS Modeler 用于在整个流中保持数据模型一致性所必需的信息。

依据数据是流入、流出还是流过节点，规范文件必需说明数据模型类型是输入、输出还是两者兼备。CLEF 节点可为传递到节点中的任何字段添加新字段，或使用程序自身生成的新字段替代传输到节点中的字段，从而影响到数据模型。规范文件中的 `OutputDataModel` 元素说明了 CLEF 节点对数据模型的影响。有关详细信息，请参阅第 66 页码第 4 章中的输出数据模型。

输入和输出文件

在运行 CLEF 节点之前，可以指定一个或多个要生成的临时文件。这些文件是**输入文件**，因为它们被输入到服务器的节点执行中。例如，模型构建器节点中可能含有模型容器，在执行节点时，该内容将被传输到指定的输入文件。有关详细信息，请参阅第 62 页码第 4 章中的输入文件。

其他临时文件是在执行服务器的节点时生成的；例如，执行模型构建器或文档构建器节点的结果。这些文件是**输出文件**，在执行节点后，它们将被传输回客户端。有关详细信息，请参阅第 63 页码第 4 章中的输出文件。

应用程序编程接口 (API)

根据您对扩展的期望，您可能需要使用应用程序编程接口 (API)。对于简单的数据转换，可能需要在规范文件中完整地定义必要的过程。但是，如果有更高级的需求，将需要具有一个或多个 API 的接口：

- CLEF 客户端 API
- CLEF 服务器端 API
- Predictive Server API (PSAPI)

CLEF **客户端 API** 是 Java API，可供需要其他控件、用户界面组件或交互（未由规范文件直接提供）的扩展使用。

CLEF **服务器端 API** 是基于 C 的 API，涵盖了多个方面，例如设置和获取执行设置、这些设置的持续时间、执行反馈、作业控制（例如中断执行）、SQL 生成和返回的对象。

Predictive Server API 是 Java API，显示了需要数据细化和预测分析功能的应用程序可以使用的 IBM® SPSS® Modeler 功能。

有关更多信息，请参阅第 9 章和程序设计第 193 页码。

文件结构

CLEF 扩展包含两组组件：

- 客户端组件
- 服务器端组件

客户端组件包含扩展规范文件、Java 类和 .jar 文件以及包含可本地化资源、图像和帮助文件的属性数据包。

服务器端组件是执行扩展节点时所需的共享库和 DLL。

客户端组件

客户端组件安装在 IBM® SPSS® Modeler 安装目录的 \ext\lib 文件夹下。服务器端组件包括：

- 规范文件
- Java 类和 .jar 文件
- 属性文件
- 图像文件
- 帮助文件

扩展文件夹

每个扩展都直接位于 \ext\lib 下自身的**扩展文件夹**中。

对扩展文件夹的建议命名规则为：

providerTag. id

其中, providerTag 是规范文件中 `ExtensionDetails` 元素的提供程序标识符, id 指相同元素的扩展标识符。

例如, 如果 `ExtensionDetails` 元素的开头为:

```
<ExtensionDetails providerTag="myco" id="sorter" ... />
```

则使用扩展文件夹名 `myco.sorter`。

规范文件

规范文件自身的名称必须为 `extension.xml`, 且必须位于扩展子文件夹的根目录下。在前面提供的示例中, 指向规范文件的路径必须在 IBM® SPSS® Modeler 安装目录下, 如下所示:

```
\ext\lib\myco.sorter\extension.xml
```

Java 类和 .jar 文件

使用客户端 Java API 的扩展中包含编译后的 Java 代码。该代码可保留为 `.class` 文件集, 也可以打包为 `.jar` 文件。

Java `.class` 文件位于相对于扩展文件夹的最上层。例如, 实现 `ActionHandler` 接口的类的路径可能为:

```
com.my_example.my_extension.MyActionHandler
```

在该示例中, `.class` 文件应当在 SPSS Modeler 安装目录的如下位置:

```
\
extension_folder\com\my_example\my_extension\MyActionHandler.class
```

`.jar` 文件可位于扩展文件夹下的任何位置。可以通过规范文件中的 `JarFile` 元素指定 `.jar` 文件的实际位置。例如, 如果扩展使用具有如下路径的 `.jar` 文件:

```
\
extension_folder\lib\common-utilities.jar
```

规范文件应当在 `Resources` 元素中包含如下条目:

```
<Resources>
  <JarFile id="util" path="lib\common-utilities.jar"/>
  ...
</Resources>
```

有关详细信息, 请参阅第 39 页码第 4 章中的 Jar 文件。

属性文件

本地化后的资源（例如，屏幕文本和错误消息及其外语翻译）可保存为扩展名为 `.properties` 的文件，该文件可位于扩展文件夹下的任意位置。有关详细信息，请参阅第 186 页码第 8 章中的属性文件。

图像和帮助文件

包含可用于显示图标图形图像的文件和包含帮助系统的文件可位于扩展文件夹下的任意位置。如果将图像文件和帮助文件分别放在其各自的子文件夹下，将会非常方便。

可以通过规范文件中 `Icon` 元素的 `imagePath` 属性声明图像文件的位置。有关详细信息，请参阅第 118 页码第 6 章中的 `Icons`。

与该方法相似，可以使用规范文件中 `HelpInfo` 元素的 `path` 属性声明帮助系统的位置。有关详细信息，请参阅第 182 页码第 7 章中的定义帮助系统的位置和类型。

示例

基于这些组件的客户端文件可能具有以下结构：

```
\ext\lib\myco.sorter
\ext\lib\myco.sorter\extension.xml
\ext\lib\myco.sorter\sorter_en.properties
\ext\lib\myco.sorter\sorter_fr.properties
\ext\lib\myco.sorter\sorter_it.properties
\ext\lib\myco.sorter\com\my_example\my_extension\MyActionHandler.class
\ext\lib\myco.sorter\help\sorter.chm
\ext\lib\myco.sorter\images\lg_sorter.gif
\ext\lib\myco.sorter\images\sm_sorter.gif
\ext\lib\myco.sorter\lib\common-utilities.jar
```

服务器端组件

执行时所需的共享库必须位于 IBM® SPSS® Modeler 安装目录中 `\ext\bin` 文件夹下的某个文件夹中，例如：

```
installation_directory\ext\bin\myco.sorter\my_lib.dll
```

请注意，共享库不得直接位于 `\ext\bin` 文件夹中。

对于 SPSS Modeler 在执行过程中直接调用的共享库，必须在规范文件的 `SharedLibrary` 元素中声明其位置。有关详细信息，请参阅第 39 页码第 4 章中的共享库。

主共享库可能要使用其他库。也可以将所有相关共享库与主共享库放置在同一位置，以方便主共享库查找相关共享库。

示例

下面是服务器端文件结构的示例：

```
\ext\bin\myco.sorter\my_lib.dll  
\ext\bin\myco.sorter\my_lib2.dll
```

节点

节点概述

创建实现新节点的扩展时，需要熟悉 IBM® SPSS® Modeler 节点的特征。这样做可帮助您在规范文件中正确定义这些节点。





SPSS Modeler 节点根据它们的功能可分为源节点、过程节点、输出节点和建模节点。在 CLEF 中，节点的分类方法略有不同。下表列出了这两个系统之间的映射：




表 2-1
CLEF 节点类型

SPSS Modeler 分类	选项板	CLEF 节点类型
源节点	源	数据读取器
过程节点	记录选项	数据变换器
	字段选项	
输出节点	图形	文档构建器
	输出（报告节点）	
	导出	数据记录器
建模节点	建模	模型构建器

创建新的 CLEF 节点时，需要将它定义为 CLEF 节点类型之一。所选的节点类型取决于该节点的主要功能。

表 2-2
节点类型和功能

CLEF 节点类型	描述	对应的节点选项板	图标形状
数据读取器	将不同格式的数据导入 SPSS Modeler。	源	
数据变换器	从 SPSS Modeler 获取数据，并以某种方式修改数据，然后将修改后的数据返回至该 SPSS Modeler 流。	记录选项，字段选项	
模型构建器	使用 SPSS Modeler 中的数据生成模型。	建模	
文档构建器	从 SPSS Modeler 中的数据生成图形或报告。	图形	

CLEF 节点类型	描述	对应的节点选项板	图标形状
		输出（报告节点）	
模型填充器 （也称为“模型块”）	为恢复到 SPSS Modeler 工作区的生成模型定义容器。	-	
数据记录器	将 SPSS Modeler 格式的数据导出为适合于其他应用程序使用的格式。	导出	

例如，在规范文件的 **Node** 元素中定义节点类型和其他属性：

```
<Node name="sort_process" type="dataTransformer"
  palette="recordOp" ... >
  -- node elements --
</Node>
```

palette 属性定义用户可以从中访问节点的 SPSS Modeler 主窗口中的选项板，此示例中定义的是“记录选项”选项板。如果省略此属性，节点会显示在“字段选项”选项板中。

IBM® SPSS® Modeler 提供了大量示例节点。有关详细信息，请参阅第 29 页码第 3 章中的关于示例。

数据读取器节点

通过数据读取器节点，可以将外部源的数据读取到 IBM® SPSS® Modeler 流中。SPSS Modeler “源”选项板上的节点与数据读取器节点等效，并且用圆形图标形状标识。

在数据读取器节点的规范中，可以包含下列内容的详细信息：

- 数据源（如文件或数据库）
- 对记录进行的所有预处理（如处理开头和结尾空格或用作记录定界符的字符）
- 是否过滤出所有记录字段
- 与每个字段关联的数据类型（如范围、集合或标志）和存储类型（如字符串、整数或实数）
- 输入数据模型是否发生了更改

数据读取器节点可以包含用于读取源数据记录的逻辑。另外，这也可以通过 SPSS Modeler 中的类型节点在下游完成。

IBM® SPSS® Modeler 提供了一个示例数据读取器节点。有关详细信息，请参阅第 29 页码第 3 章中的关于示例。

数据变换器节点

数据变换器节点从 IBM® SPSS® Modeler 流获取数据，并以某种方式修改数据，然后将修改后的数据返回至该流。SPSS Modeler “记录选项”和“字段选项”选项板上的节点都是数据变换器节点，并且用六边形图标形状标识。

在数据变换器节点的规范中，可以包含下列内容的详细信息：

- 变换哪些记录或字段
- 修改数据的方式

IBM® SPSS® Modeler 提供了一个示例数据变换器节点。有关详细信息，请参阅第 29 页码第 3 章中的关于示例。

模型构建器节点

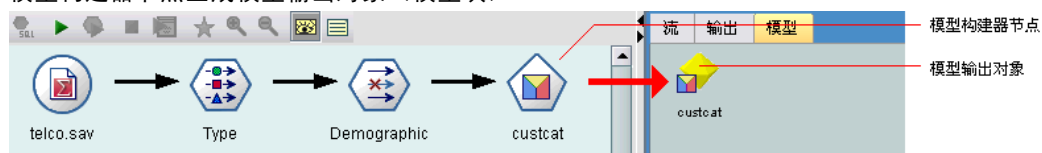
有关在 IBM® SPSS® Modeler 中构建模型的概述，请参阅《SPSS Modeler 15 应用程序指南》中的“建模简介”。

模型构建器节点生成的对象会显示在 SPSS Modeler 主窗口中管理器窗格的“模型”或“输出”选项卡上。

SPSS Modeler “建模”选项板上的节点是模型构建器节点的示例，并且用五边形图标形状标识。

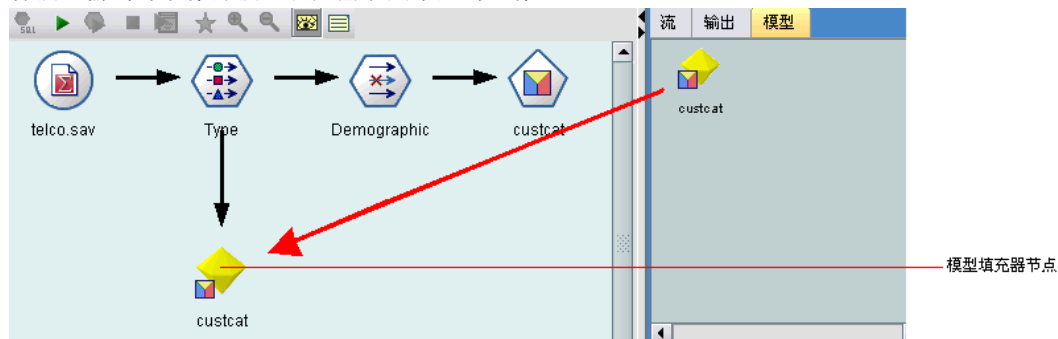
执行后，模型构建器节点会在“模型”选项卡上生成**模型输出对象**（也称为“模型块”）。

图片 2-1
模型构建器节点生成模型输出对象（模型块）



将生成的模型添加到工作区时，它将采取模型填充器节点的形式。

图片 2-2
将模型输出对象作为模型填充器节点添加到工作区



在模型构建器节点的规范中，可以包含下列内容：

- 模型构建详细信息，如用于生成模型的算法以及哪些输入和输出字段可用于对模型的数据进行计分
- 模型使用的属性
- 用于容纳输出对象的容器
- 节点对话框的用户界面
- 执行节点时使用的属性和文件
- 执行节点对输入数据模型的影响
- 执行节点时生成的模型输出对象以及任何其他对象的标识符
- 模型填充器节点的标识符（请参阅[模型填充器节点第 13 页码](#)）

注意：定义模型构建器节点时，需要在相同规范文件的其他位置定义实际的模型输出对象和模型填充器节点。

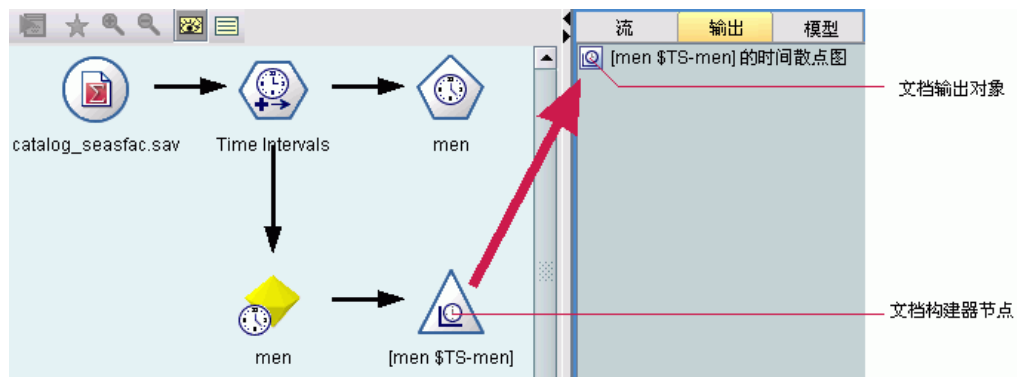
IBM® SPSS® Modeler 提供了一个示例模型构建器节点。 [有关详细信息，请参阅第 29 页码第 3 章中的关于示例。](#)

文档构建器节点

文档构建器节点生成的对象会显示在 IBM® SPSS® Modeler 主窗口中管理器窗格的“输出”选项卡上。“图形”选项板上的节点是文档构建器节点的示例，并且用三角形图标形状标识。

执行后，文档构建器节点会在管理器窗格的“输出”选项卡上生成**文档输出对象**。

图片 2-3
文档构建器节点生成文档输出对象



与模型输出对象相反，不能将文档输出对象添加回 SPSS Modeler 工作区。

在文档构建器节点的规范中，可以包含下列内容：

- 文档构建详细信息，如要包含文档生成控件的节点对话框选项卡
- 文档使用的属性
- 用于容纳输出对象的容器
- 节点对话框的用户界面

- 执行节点时使用的属性和文件
- 执行节点时生成的文档输出对象和任何其他对象的标识符

注意：定义文档构建器节点时，需要在相同规范文件的其他位置定义实际的文档输出对象。

模型填充器节点

模型填充器节点定义用于容纳生成模型的容器，将生成的模型从管理器窗格的“模型”选项卡添加到 IBM® SPSS® Modeler 工作区时使用该容器。

在模型填充器节点的规范中，可以包含下列内容的详细信息：

- 容纳模型的容器（如果可以如文本和 HTML 等多种格式生成模型输出，则为多个容器）
- 用户浏览“模型”选项卡上的填充器节点或在工作区打开填充器节点时显示的对话框的用户界面详细信息
- 输出数据模型
- 执行包含节点的流时所执行的处理
- 用于处理执行包含节点的流时所生成的对象的构造函数

数据记录器节点

数据记录器节点将 IBM® SPSS® Modeler 格式的数据导出为适合于其他应用程序使用的格式。SPSS Modeler “导出”选项板上的节点是数据记录器节点，并且用矩形图标形状标识。

在数据记录器节点的规范中，可以包含下列内容：

- 流数据将被写入的文件或数据库的详细信息
- 是否发布整个流以使它可以嵌入到外部应用程序中（可选）

菜单、工具栏和选项板

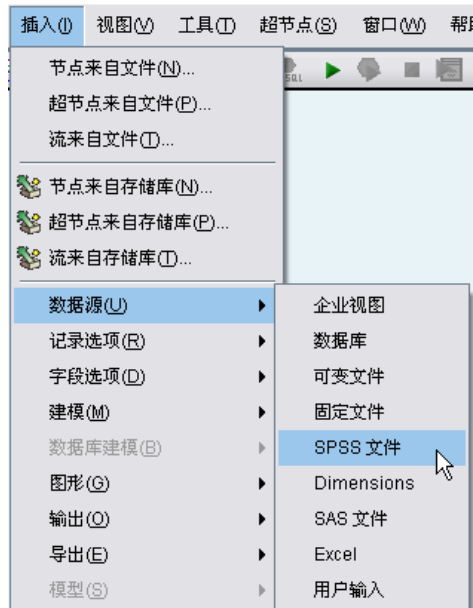
用户可以从 IBM® SPSS® Modeler 菜单、工具栏或选项板访问扩展。扩展可以实现节点或执行指定的操作。

也可从工具栏访问可以从明确指定的菜单进行访问的扩展（节点或操作），反之亦然。可以从“插入”菜单上的相应项目自动访问可从选项板访问的节点。

菜单和子菜单

用户可以从“插入”菜单访问标准的 IBM® SPSS® Modeler 节点。此菜单最后一组中的每一项（“模型”除外）均有一个子菜单，用于提供对相关节点集访问。

图片 2-4
插入菜单



这些项直接与节点选项板上的条目相对应。将节点添加到选项板时，也会自动将其添加到“插入”菜单的相应组中。

如果扩展定义的操作无法通过节点访问，则可以通过添加下列一项或多项使该扩展可用：

- 系统菜单或子菜单的新项
- SPSS Modeler 的新菜单
- 工具栏的新项（请参阅[工具栏第 14 页码](#)）

新菜单或菜单项还可以显示与扩展关联的图标，如与“插入”菜单项中的某些项一样。

有关更多信息，请参阅[菜单第 120 页码](#)和[菜单项第 121 页码](#)。

工具栏

如果扩展定义的操作无法通过节点来访问，则可以通过将其添加到 IBM® SPSS® Modeler 的主工具栏来使该扩展成为可用。

图片 2-5
将项添加到主工具栏



在这种情况下，建议隐藏该操作的标签。

也可以将项添加到节点对话框或输出窗口的工具栏中。可以选择显示或隐藏项标签，如下面的示例所示：

图片 2-6
将项添加到对话框工具栏中



有关详细信息，请参阅第 123 页码第 6 章中的工具栏项。

选项板和子选项板

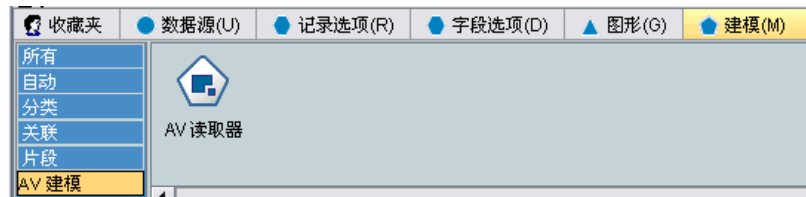
如果扩展定义了新节点，则可以将该节点放到一个标准的 IBM® SPSS® Modeler 选项板或子选项板的任意位置。

图片 2-7
标准选项板上的新节点



可以向标准子选项板添加一个条目，并可以从此处访问该节点。

图片 2-8
自定义添加到标准子选项板上的新节点



可以定义自定义选项板，并将新节点放入其中。

图片 2-9
自定义选项板上的新节点



自定义选项板可以包含自定义子选项板。

图片 2-10
自定义选项板的自定义子选项板上的新节点



有关更多信息，请参阅 [节点第 54 页码](#) 和 [用户界面（选项板）部分第 48 页码](#)。

设计节点图标

对于在 CLEF 中创建的每一个新节点，均可以为在屏幕上标识该节点的图标提供一个中央图像。

注意：也可以不提供图像，如果不指定图像，则显示 IBM® SPSS® Modeler 提供的默认图像（这在开始开发节点时非常有用）。

图片 2-11
CLEF 图标的默认图像

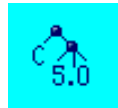


标准 SPSS Modeler 图标由以下三层组成：

- 边框
- 背景
- 中央图像








对于新节点，您只需提供中央图像（也称为**轮廓**），SPSS Modeler 会对边框和背景进行处理。轮廓图像需要透明背景，以免遮挡图标背景层。在本节中，表示轮廓时采用的是彩色背景以便指示透明度。

图片 2-12
包含指示透明度的彩色背景的轮廓层



这是典型的 SPSS Modeler 建模图标的组成方式。

表 2-3
节点图标和已生成模型图标的组成




	节点图标	生成的模型图标
边框		None
背景		
轮廓		
图像显示方式		

边框

节点的功能由图标边框的形状表示。 [有关详细信息，请参阅第 9 页码节点概述。](#)

如果节点启用了缓存，边框形状上会添加一个文档符号缩图。节点上的白色文档图标指示其高速缓存为空。当高速缓存已满时，该文档图标将保持绿色不变。

表 2-4
节点边框和缓存状态

缓存状态	示例
无缓存	
启用缓存	
缓存已满	

该系统提供了不同的边框符号，并且 IBM® SPSS® Modeler 会进行必要的处理以在正确的时间显示正确的符号。

背景

除生成的模型节点和模型填充器节点的图标以外，其他节点图标的背景均会更改颜色以指示状态。

表 2-5
节点背景

状态	颜色	示例
未选中的	灰色	
Selected	蓝色	
错误	红色	
数据库中执行的操作	紫色	

另外，该系统还提供了背景图像，并且 IBM® SPSS® Modeler 会进行必要的处理以在所有情况下都能显示正确的背景。

图形要求

对于每个新的 CLEF 节点，均可创建下列版本的轮廓层图像：

- 大尺寸（49 x 49 像素）图像，针对流工作区上的节点
- 小尺寸（38 x 38 像素）图像，针对位于屏幕底部选项板管理器中的节点

如果要在菜单、工具栏、浏览器或输出窗口的标题栏中显示图标，则还需创建：

- 缩图尺寸（16 x 16 像素）图像

如果节点生成模型，则还需创建：

- 小尺寸（38 x 38 像素）图像，设计为移到左下角以覆盖生成的模型图标（金色块）

注意：大于这些尺寸的图像在 IBM® SPSS® Modeler 中显示时将被剪裁。

有关详细信息，请参阅第 118 页码第 6 章中的 Icons。

创建自定义图像

为节点创建的图像应传达该节点的主要功能。考虑到国际用户，请注意使用的图像不要特定于某一个国家/地区，并且不会引起其他国家/地区用户的误解。

要创建用于 CLEF 的自定义图像：

- ▶ 通过使用支持透明度的图形包将绘图工作区设置为适当的尺寸并绘制图像版本。
- ▶ 将每个版本（大尺寸、小尺寸等）保存为具有下列特征的不同 .gif 文件：
 - 透明背景
 - 16 色（4 位）或更高的颜色深度

将图像背景设为透明的方法取决于使用的图形包。例如，您可能可以将背景颜色直接设置为透明，或者可能需要先指定一种透明颜色，然后再用此颜色“描画”图像背景。

对于图像文件，建议采用下列 IBM® SPSS® Modeler 内部使用的文件命名规则：

表 2-6
图像文件命名规则

图像类型	文件名
Large	lg_node.gif
小(S)	sm_node.gif
缩图尺寸	node16.gif
生成的模型	sm_gm_node.gif

- ▶ 通过在规范文件中引用图像文件（请参阅[将图像文件添加至节点规范](#)）并将新节点添加至 SPSS Modeler（请参阅[测试 CLEF 扩展第 219 页码](#)）来测试图像的外观。

将图像文件添加至节点规范

创建图像文件后，将它们复制到将要运行 IBM® SPSS® Modeler 的计算机上的文件夹中。在规范文件中，需要声明图像相对于 SPSS Modeler 安装目录中 \ext\lib\provider.nodename 文件夹的路径，因此应将这些文件部署到易于查找的文件夹。[有关详细信息，请参阅第 118 页码第 6 章中的 Icons。](#)

在规范文件中，可以通过 **Node 规范 UserInterface** 部分中的 **Icons** 元素将大图标图形和小图标图形文件与自定义节点关联起来，例如：

```
<Icons>
  <Icon type="standardNode" imagePath="images/lg_mynode.gif" />
  <Icon type="smallNode" imagePath="images/sm_mynode.gif" />
</Icons>
```

对于模型构建器或文档构建器节点，还需要在 **ModelOutput** 规范（对于模型构建器节点）或 **DocumentOutput** 规范（对于文档构建器节点）的 **UserInterface** 部分中引用缩图（16 x 16 像素）版本，例如：

```
<Icons>
  <Icon type="standardWindow" imagePath="images/mynode16.gif" />
</Icons>
```

对于模型填充器节点，还需要在 **Node 规范** 的 **UserInterface** 部分中引用生成的模型版本，例如：

```
<Icons>
  <Icon type="standardNode" imagePath="images/lg_gm_mynode.gif" />
  <Icon type="smallNode" imagePath="images/sm_gm_mynode.gif" />
</Icons>
```

设计对话框

本节介绍标准 IBM® SPSS® Modeler 节点对话框的特征，以帮助您设计一致的对话框。

关于节点对话框

节点对话框可为最终用户提供一个可修改执行设置的界面。对话框的外观很重要，因为要在这里更正和修改节点行为。该界面必须包含所有必要的信息，而且要易于使用。

节点行为是通过使用各种基于对话框的**控件**来进行更改的，这些控件是用户可以进行交互的用户界面元素。对话框可以包含许多控件，如单选按钮、复选框、文本框和菜单。CLEF 提供了各种各样的控件，您可以将它们用于您的对话框设计。[有关详细信息，请参阅第 134 页码第 6 章中的属性控件规范。](#)

图片 2-13
含有单选按钮、复选框、组合框和选项卡的对话框示例



由控件修改的参数类型决定出现在对话框中的控件，其中有些类型可提供可选控件。可以通过规范文件中的 **Tab** 元素对新选项卡上选项进行分组。[有关详细信息，请参阅第 24 页码选项卡区域。](#)

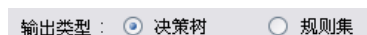
注意：即使未指定扩展要执行的处理，也可以测试该扩展的用户界面外观。[有关详细信息，请参阅第 219 页码第 10 章中的测试 CLEF 扩展。](#)

对话框的设计原则

定义对话框的控件时，请考虑下列原则：

- 仔细考虑控件显示标签上要使用的文本。该文本应简短适度，同时能够传达准确的信息。如果设计面向国际市场，请注意翻译后的文本长度可能会与原始文本长度有显著差异。
- 对参数使用合适的控件。例如，对于仅有两个值的参数，最佳选择并不一定总是复选框。IBM® SPSS® Modeler C5.0 节点对话框使用单选按钮来使用户选择具有一决策树和一规则集这两个值的输出类型。

图片 2-14
在两个值之间进行选择



此设置可以用一个标签为**决策树**的复选框来表示。选中时，输出类型为决策树；取消选中时，输出则为规则集。虽然结果一样，但在本例中使用单选按钮可使用户更易于理解选项。

- 文件名控件通常位于对话框的顶部。
- 组成节点重要部分的控件置于对话框的上端。例如，图形节点显示数据中的字段。选择这些字段是该对话框的主要功能，因此字段参数位于顶部。
- 复选框和单选按钮通常允许用户选择需要进一步信息的选项。例如，在 C5.0 对话框中选择使用**提高**时，需要在分析中包含一个表示**尝试次数**的数字。

图片 2-15
含有关联控件的复选框



进一步信息总是放置在选项选择之后，要么在其右边，要么在其正下方。

CLEF 对话框就像标准 SPSS Modeler 对话框一样使用 SPSS Modeler 的提交编辑方式：仅在用户单击**确定**、**应用**或**执行**终端节点之后，才将对话框中显示的值复制到节点。同样，只有用户取消或重新显示对话框或单击**刷新**按钮后，对话框显示的信息才会被更新（例如，在当前节点的上游进行操作，导致节点的输入字段被更改时）。

对话框组件

对话框可包含下列组件：

- 标题栏
- 图标区域
- 工具栏和菜单区域可包含：
 - 文件、生成、视图、预览、刷新以及其他按钮（取决于节点）
 - 最大化/常规尺寸按钮
 - 帮助按钮

- 状态区域
- 面板区域
- 选项卡区域
- 按钮区域

每一个自定义节点均需要一个对话框，用户打开该节点时会显示该对话框。假设您的规范文件包含一个 **Node** 元素，且该元素的 **UserInterface** 部分包含一个 **Tabs** 元素，当您打开该节点时会看到上述所有的对话框组件。根据节点类型的不同，选项卡区域和按钮区域至少包含下列内容：

表 2-7
不同类型的节点至少包含的选项卡区域和按钮区域的内容

节点类型	Tabs	按钮
数据读取器	注解（工具栏区域含有“刷新”按钮）	确定、取消、应用和重置
数据变换器	注解	确定、取消、应用和重置
数据记录器	发布和注解	确定、取消、执行、应用和重置
模型构建器	注解	确定、取消、执行、应用和重置
文档构建器	注解	确定、取消、执行、应用和重置
模型填充器	汇总和注解	确定、取消、应用和重置

首先定位节点对话框，这样当用户打开该节点时，节点图标会添加到它所表示的节点上。用户可以移动对话框，但下次打开该节点时对话框并不会显示在这个新位置上。如果用户移动了对话框，并且随后另一对话框部分或完全遮住了它，则可以在工作区中双击原始节点以使第一个对话框再次显示在前面。对话框是无模式的（即相同的用户输入始终会产生相同的操作）并可以调整大小。

对话框中所有可编辑的字段均支持下列键盘快捷键：

快捷键	作用
Ctrl-C	粘贴
Ctrl-V	粘贴
Ctrl-X	剪切

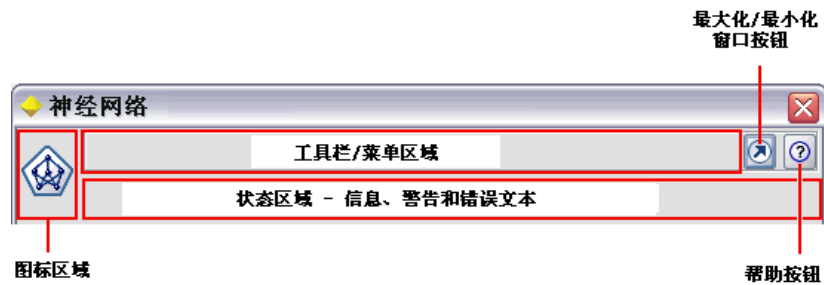
标题栏

节点对话框的标题栏包含缩图版的 IBM® SPSS® Modeler 金色块图标，紧随其后的是模型名称。文本来自模型名称控件的设置。默认还在右上角提供“关闭”按钮 (X)。

图标区域

节点图标显示在对话框左上角附近的图标区域。

图片 2-16
节点对话框顶部



这是还可用于主窗口底部的节点选项板的小尺寸 (38 x 38 像素) 版本的图标，而不是显示在工作区中的大版本图标。

注意：标题栏左端的缩图金色块图标已硬编码到所有节点对话框中。

工具栏和菜单区域

对话框的最顶部区域保留为工具栏和菜单区域。

数据读取器和数据变换器节点对话框在此区域具有“预览”按钮，显示输入数据的样本。

数据读取器节点对话框还具有“刷新”按钮，它将更新节点显示的信息（例如，当节点的输入字段发生更改时）。

模型填充器节点拥有文件、生成和视图菜单按钮，使用户能够执行各种操作，如导出模型或生成新节点。模型填充器节点还有“预览”按钮，但在本例中显示输入数据的样本，以及应用节点时所创建的其他列。

此区域的右侧包含每一个节点对话框均有的两个按钮：

- 最大化/常规尺寸按钮
- “帮助”按钮

最大化/常规尺寸按钮

此按钮可以将对话框调整为全屏尺寸。再次使用此按钮时可将对对话框收缩回最大化之前的尺寸。

帮助按钮

此按钮可打开节点的上下文相关的帮助。对于具有选项卡的对话框或输出窗口，会显示该选项卡的帮助。还可以用 F1 键来访问帮助。

状态区域

对话框顶部的剩余区域保留为显示信息、警告或错误文本。源节点会在此显示源数据文件的完整路径和文件名。各个节点可在此区域显示其他节点特定的信息。为此区域指定的任何文本均应限制在两行内。

面板区域

这是对话框的主要区域，包含节点的所有控件和显示区域。每一个选项卡均可以有不同的面板区域。每一个面板可以是下列类型之一：

- 文本浏览器
- 扩展对象
- Properties

还可以指定子面板，子面板是在新窗口中打开的单独对话框，并通过面板上的操作按钮进行调用。

[有关详细信息，请参阅第 127 页码第 6 章中的面板规范。](#)

选项卡区域

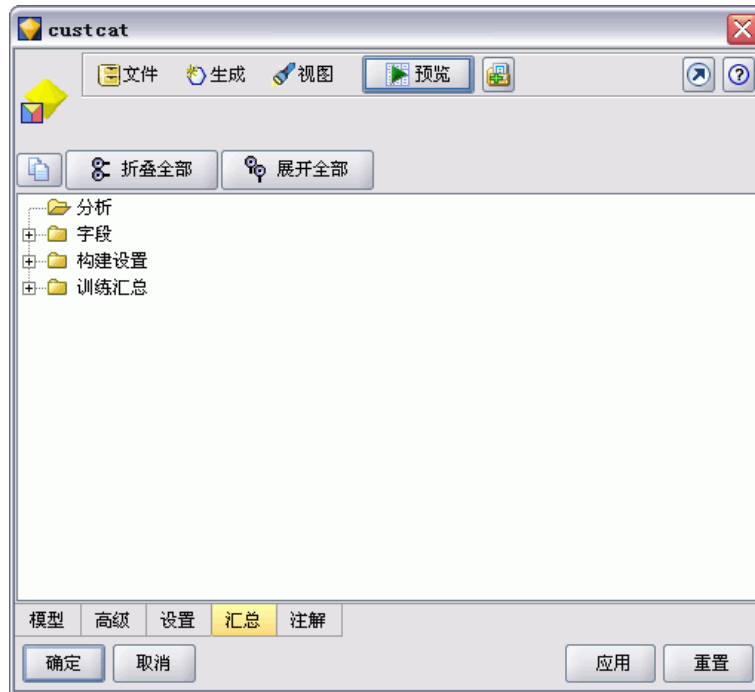
节点对话框可以包含下列选项卡：

- 一个或多个用户提供的节点特定的选项卡
- 一个“汇总”选项卡（仅适用于模型输出对象和模型填充器节点）
- 一个“注解”选项卡

节点特定的选项卡定义在 CLEF 规范文件的 **Tabs** 部分中。[有关详细信息，请参阅第 124 页码第 6 章中的 Tabs。](#)

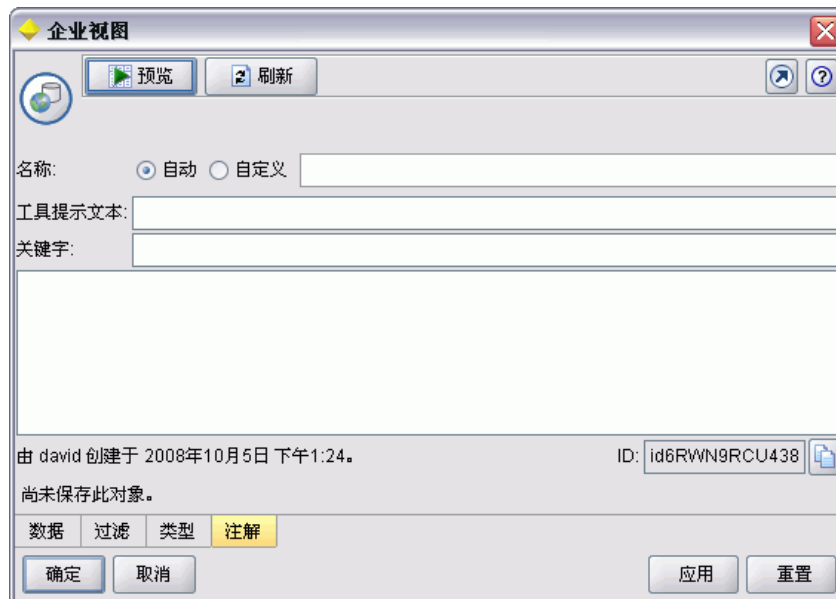
模型输出对象和模型填充器节点的对话框均包含一个系统提供的“汇总”选项卡。该选项卡显示有关生成模型的汇总信息，包括使用的字段、构建设置和模型评估过程。结果以树状视图显示，通过单击指定项可以扩展或合并树状视图。

图片 2-17
“汇总”选项卡



系统为所有节点对话框提供了“注解”选项卡，用户可以通过该选项卡指定节点的相关信息。该选项卡包含节点名称、工具提示文本和一个较长的注释字段。

图片 2-18
“注解”选项卡



名称。默认节点名称在规范文件的 Node 元素的 Label 属性中指定（请参阅[节点第 54 页码](#)）。用户可以重命名节点，方法是选择自定义，在“自定义”编辑字段中输入一个名称，然后单击应用或确定。虽然通过选择自动可恢复默认名称，但新名称会跨各个会话保留。在“注解”选项卡上指定的自定义名称会替代对话框中任何其他选项卡上指定的自定义名称。

工具提示文本。此处指定的文本显示为工作区上节点的工具提示。如果此处未指定工具提示文本，当用户将光标悬停在此节点上时不会显示工具提示。

关键字。用户可以指定要用于工程报告的关键字，以及搜索或跟踪 IBM® SPSS® Collaboration and Deployment Services Repository 中存储的对象时使用的关键字。

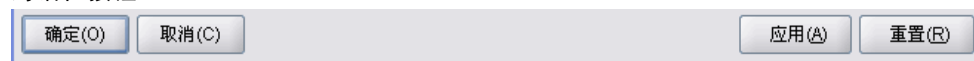
注释面板。用户可在此区域输入注释文本。

创建和保存信息。这是一个不可编辑的文本区域，它显示已存文件的创建信息、名称以及日期/时间（日期/时间格式因区域不同而有所不同）。如果未进行保存，此字段将显示“该项目尚未保存”。

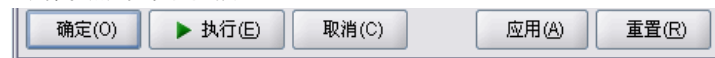
按钮区域

每一个对话框的底部都会显示应用、重置、确定和取消按钮。如果节点是终端节点（处理流数据的可执行节点），则还会显示执行按钮。

图片 2-19
对话框按钮



图片 2-20
终端节点的对话框按钮



确定。应用所有设置并关闭对话框。当第一次从节点打开对话框时，该按钮具有焦点（由按钮周围的蓝色矩形表示），按 Enter 键也可以执行“确定”操作。

取消。关闭对话框并保留打开对话框之前或上一次执行“应用”操作之后的设置。当整个对话框具有焦点时，按 Esc 键也可以执行“取消”操作。

执行。应用所有设置，关闭对话框并执行终端节点。

应用。保存对话框的设置，以便下游操作可以使用这些设置。

重置。将整个对话框重置为打开对话框时或上一次执行“应用”操作之后的值。

设计输出窗口

本节介绍标准 IBM® SPSS® Modeler 输出窗口的特征，以帮助您在 CLEF 中设计一致的输出窗口。

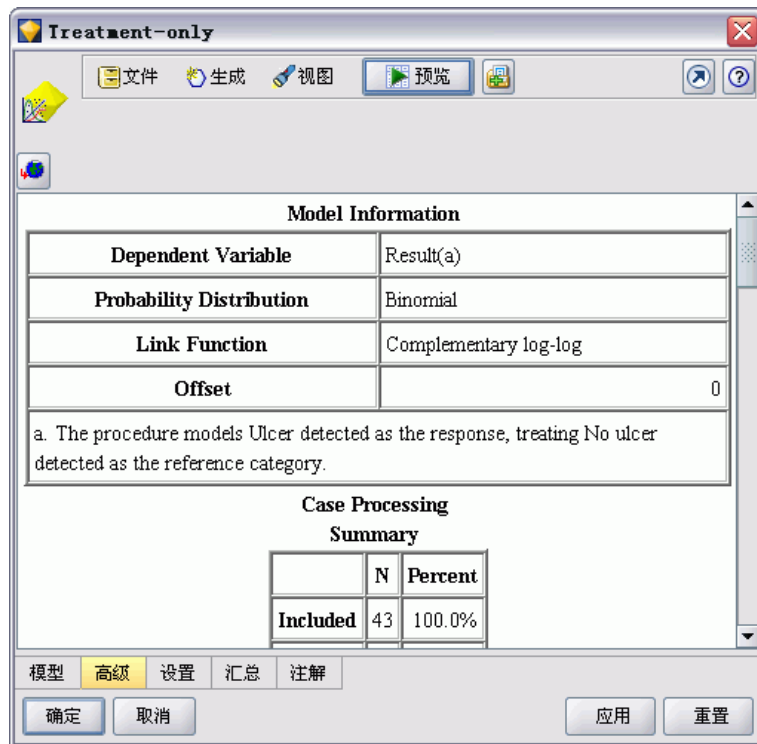
输出窗口显示的输出可以来源于：

- 模型，如对一组数据计分的模型（将模型应用于一组数据）
- 文档，如图形或报告

有关详细信息，请参阅第 113 页码第 6 章中的关于用户界面。

典型的输出窗口如下所示：

图片 2-21
模型输出窗口



输出窗口与节点对话框相似，但有下列不同之处：

- 标题栏具有节点特定的缩图图标，而不是通用的金色块图标
- 省略了主节点图标
- 在工具栏和菜单区域中，省略了最大化/常规按钮（在文档输出窗口中可由“关闭”和“删除”按钮代替），但仍然可以通过鼠标调整窗口大小
- 省略了状态区域
- 选项卡通常为：
 - “模型”选项卡（用于模型输出窗口）显示预测变量重要性数据（如果该选项在模型节点上选中）
 - 用于输出的单个选项卡
 - “汇总”选项卡（用于模型输出窗口）显示模型的汇总详细信息
 - “注解”选项卡（注解值来自生成输出的节点）
- 按钮区域有“确定”、“取消”、“应用”和“重置”按钮

CLEF 提供了与前面一个所示插图相类似的默认模型输出窗口和文档输出窗口。在规范文件中使用 `ModelOutput` 或 `DocumentOutput` 元素时，通常会显示这两个窗口。有关详细信息，请参阅第 54 页码第 4 章中的对象标识符。

另外，可以通过指定 `ModelOutput` 或 `DocumentOutput` 元素，使用自己设计的自定义窗口完全替换默认的输出窗口。有关详细信息，请参阅第 178 页码第 6 章中的自定义输出窗口。

CLEF 示例

关于示例

为帮助您熟悉 CLEF，IBM® SPSS® Modeler 安装中包含了一组示例节点及其完整的源代码。这些基本节点功能有限，旨在帮助您理解 CLEF 的工作原理及其用法。您可以马上或随时测试这些节点。

示例包括：

- 数据读取器节点（名为 Apache Log Reader）
- 数据转换器节点（名为 URL Parser）
- 文档构建器节点（名为 Web Status Report）
- 模型构建器节点（名为 Interaction）

在使用这些示例之前，必须先将其激活。

激活示例

这些示例作为 IBM® SPSS® Modeler 安装的一部分以压缩格式安装到 Demos 目录中。您需要激活这些示例，方法是将这些文件提取到如下所示的正确位置。

在安装 SPSS Modeler 的计算机上：

- ▶ 如果 SPSS Modeler 正在运行，请退出。
- ▶ 在 SPSS Modeler 安装的 Demos 文件夹中找到文件 clef_examples_ext_lib.zip。
- ▶ 将 clef_examples_ext_lib.zip 的内容提取到 SPSS Modeler 安装目录的 \ext\lib 文件夹中。

要仅安装 SPSS Modeler：

- ▶ 将 clef_examples_ext_bin.zip 的内容提取到 SPSS Modeler 安装目录的 \ext\bin 文件夹中。

要分别对 SPSS Modeler 和 IBM® SPSS® Modeler Server 安装进行配置：

- ▶ 将 clef_examples_ext_bin.zip 的内容提取到 SPSS Modeler 和 SPSS Modeler Server 安装目录的 \ext\bin 文件夹中。

- ▶ (仅适用于 UNIX 和 64 位 Windows 服务器) 除上述步骤之外, 还需要使用 clef_examples_ext_bin.zip 中提供的 makefile 编译示例的源代码。 [有关详细信息, 请参阅第 33 页码检查源代码。](#)

最后, 在所有情况下:

- ▶ 启动 SPSS Modeler 并确保节点选项板上显示如下节点:

选项板选项卡	节点
源	Apache Log Reader
字段选项	URL Parser
建模	交互
输出	Web Status Report

数据读取器节点 (Apache Log Reader)

数据读取器节点示例是从 Apache HTTP Web 服务器的访问日志文件中读取数据的源节点。访问日志中包含了 Web 服务器已处理的所有请求的详细信息。日志记录的格式为综合日志格式, 例如:

```
IP_address -- [09/Jul/2007:07:57:38 +0000] "GET /lsearch.php?county_id=3 HTTP/1.1" 200 16348
"http://www.google.co.uk/search?q=thunderbirds+cliveden&hl=en&start=10&sa=N"
"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)"
```

您可以使用该示例节点将日志记录转换为易于读取的表格格式。

要使用 Apache Log Reader 节点:

- ▶ 如果尚未激活 CLEF 示例, 请立即激活。 [有关详细信息, 请参阅第 29 页码激活示例。](#)
- ▶ 打开 IBM® SPSS® Modeler。
- ▶ 在节点选项板的“源”选项卡中选择 **Apache Log Reader**, 并将该节点添加到工作区中。
- ▶ 编辑节点。在“选项”选项卡的“Apache 日志文件”字段中, 输入:

```
demos_folder\combined_log_format.txt
```

其中 demos_folder 是指 Demos 文件夹在 IBM® SPSS® Modeler 安装目录中的位置 (不要使用 \$CLEO_DEMOS 格式)。

- ▶ 单击**确定**。
- ▶ 为流添加类型节点。
- ▶ 编辑类型节点。单击**读取值**以读取数据, 然后单击**确定**。
- ▶ 将表节点附加到类型节点并执行流。以表格格式显示日志文件的内容。
- ▶ 保存流, 以供下面的两个示例使用。

数据转换器节点 (URL Parser)

数据转换器节点示例用于进一步处理前一示例返回的数据。选择一个 ID 字段（该字段对每一行都应当包含唯一值）和一个包含 URL 的输入字段。该节点会生成由这两个字段组成输出以及另外解析到各个生成字段的 URL 数据。例如，如果 URL 记录包含如下查询字符串：

```
http://www.dummydomain.co.uk/resource.php?res_id=89
```

该记录将被解析为：

已生成字段	内容
URLfield_server	http://www.dummydomain.co.uk
URLfield_path	/resource.php
URLfield_field	res_id
URLfield_value	89

要使用 URL Parser 节点：

- ▶ 如果前一示例中的流已经关闭，请立即将其打开。该流中包含了 Apache Log Reader 和类型节点。
- ▶ 在节点选项板的“字段选项”选项卡中，将 URL Parser 节点附加到类型节点。
- ▶ 编辑 URL Parser 节点。从“ID 字段”下拉列表中，选择 ReturnedContentSize。从“URL 字段”下拉列表中，选择 ReferralURL。单击确定。
- ▶ 将表节点附加到 URL Parser 节点并执行流。此时将显示 ReturnedContentSize 和 ReferralURL 字段，此外，ReferralURL 还将解析到四个单独的生成字段：ReferralURL_server、ReferralURL_path、ReferralURL_field 和 ReferralURL_value。

文档构建器节点 (Web Status Report)

文档构建器节点示例会读取继承自 Web 服务器日志的数据，并生成 HTML 文件形式的报告。该报告中包含的表反映了返回各种 HTTP 状态代码（例如，200、302、404 等）的日志记录的百分比。

要使用 Web Status Report 节点：

- ▶ 如果第一个示例中的流已经关闭，请立即将其打开。该流包含了 Apache Log Reader 和类型节点。如果您的流具有第二个示例中的 URL Parser 节点，在本示例中将忽略该节点。
- ▶ 在节点选项板的“输出”选项卡中，将 Web Status Report 节点附加到类型节点。
- ▶ 编辑 Web Status Report 节点。从“状态代码字段”下拉列表中，选择 StatusCode。单击执行。此时将显示包含报告内容的输出窗口。

模型构建器节点 (Interaction)

模型构建器节点示例的操作与其他示例无关，您可以用它以标准（非交互）方法构建简单的模型，也可以在生成该节点之前将其与模型交互。该模型可尝试预测电信公司的客户流失情况。

要使用 Interaction 节点：

- ▶ 如果尚未激活 CLEF 示例，请立即激活。 [有关详细信息，请参阅第 29 页码激活示例。](#)
- ▶ 在 IBM® SPSS® Modeler 中创建新流。
- ▶ 添加可从 Demos 目录导入文件 telco.sav 的 Statistics 文件源节点。
- ▶ 在“类型”选项卡中，单击**读取值**，然后在消息框中单击**确定**以确认。
- ▶ 将流失字段（列表中的最后一个）的角色设置为**目标**，然后单击**确定**。
- ▶ 在节点选项板的“建模”选项卡中，将交互节点附加到源节点。

要测试标准（非交互）的模型构建：

- ▶ 运行流以在流和屏幕右上角的“模型”选项板中创建模型块。
- ▶ 将表节点附加到模型块。
- ▶ 执行表节点。滚动到表输出窗口的右侧，以查看流失情况的预测。字段 `$I-churn` 中包含了预测值，而 `$IP-churn` 则显示了这些预测的置信度值（从 0.0 到 0.1）。

要测试交互的模型构建：

- ▶ 在“交互模型构建器”对话框的“模型”选项卡中，选择**启动交互会话**。
- ▶ 单击**执行**可显示“交互测试”对话框。
- ▶ 在“交互测试”对话框中，单击**启动构建任务**以显示模型构建进程。
- ▶ 完成模型构建操作后，在对话框中选择已添加到“构建任务”表中的行。
- ▶ 在对话框顶部的工具栏区域中，单击带有黄色菱形图标的按钮。该操作将在屏幕右上角的“模型”选项板中生成模型输出对象（名为 `model_1`）。

除非该交互生成的模型使用其他名称，否则它与第一个模型的名称相同。重复从单击**启动构建任务**开始的过程将生成另一个名为 `model_2` 的相同模型，以此类推。

检查规范文件

了解 CLEF 工作原理的一个好办法是检查所提供示例的规范文件。这些文件位于：

```
install_dir\ext\lib\  
extension_folder\extension.xml
```

其中 `install_dir` 是 IBM® SPSS® Modeler 安装目录，`extension_folder` 是指下列内容之一：

- spss.apachelogreader
- spss.interaction
- spss.urlparser
- spss.webstatusreport

您可能会看到 `\ext\lib` 下列出的其他扩展文件夹，这些文件夹与使用 CLEF 生成的系统提供的 IBM® SPSS® Modeler 节点有关。这些节点是否在安装中显示取决于许可的 SPSS Modeler 模块。您可能还会发现浏览这些节点的规范文件很受启发，但是**不要以任何方式更改这些文件**。如果更改了这些文件，这些节点将无法正常工作，在这种情况下，您必须重新安装相关的 SPSS Modeler 产品。IBM Corp. 不支持对系统提供的文件进行更改

检查源代码

为了便于引用，我们还提供了示例节点的完整源代码。所有示例代码都使用 C++ 服务器端库，但 Interaction 节点还使用客户端 Java 类。

在激活示例时，会自动提取和安装源代码文件，如下所示：

位置	内容
...\ext\lib\spss.interaction\src	父文件夹中 ui.jar 文件中的 .class 文件的 Java 源代码
...\ext\bin\spss.apachelogreader\src ...\ext\bin\spss.interaction\src ...\ext\bin\spss.urlparser\src ...\ext\bin\spss.webstatusreport\src	父文件夹中 DLL 的 C++ 源代码和项目文件

删除示例

如果无需再查看 IBM® SPSS® Modeler 中的示例节点，则可以将其删除，操作步骤如下所示：

- ▶ 退出 IBM® SPSS® Modeler。
- ▶ 从 SPSS Modeler 安装的 `\ext\bin` 和 `\ext\lib` 目录中删除示例文件夹。不要误删任何标准 SPSS Modeler 文件夹。如果误删，必须重新安装相关的 SPSS Modeler 产品。要删除的文件夹包括：
 - spss.apachelogreader
 - spss.urlparser
 - spss.webstatusreport
 - spss.interaction

下次启动 SPSS Modeler 时，这些更改才生效。

规范文件

规范文件概述

每个 CLEF 扩展都必须包含一个用于定义所有扩展特征的 XML 文件。这个文件称为**规范文件**，通常名为 `extension.xml`。规范文件包括以下几个部分：

- **XML 声明**。（可选）XML 版本的声明及其他信息。
- **扩展元素**。文件的主要部分；包含下列所有部分。
- **扩展详细信息部分**。用于指定扩展的相关基本信息。
- **资源部分**。用于指定扩展生效所需的外部资源，如资源数据包、JAR 文件和共享库。
- **通用对象部分**。（可选）定义在扩展中可能会使用或被其他对象引用的项目，如模型、文档和属性类型。
- **用户界面（选项板）部分**。（可选）定义在其中显示节点的自定义选项板或子选项板。
- **对象定义部分**。标识由扩展定义的对象，如节点、模型输出和文档输出。

每个部分既可以包含静态声明（如元素中的组件），也可以只包含动态过程（如节点的输出数据模型的计算过程），或者两者都包含。完整的 CLEF 规范文件格式如下所示：

```
<?xml version="1.0" encoding="UTF-8" ?>
<Extension ...>
  <ExtensionDetails .../>
  <Resources
    Resources section
  </Resources>
  <CommonObjects>
Common Objects section
  </CommonObjects>
  <UserInterface>
User Interface (Palettes) section
  </UserInterface>
  Object Definition section
    object definition
    object definition
    object definition
    ...
</Extension>
```

注释行

在规范文件的任意位置，您都可以添加如下格式的注释行：

```
<!-- comment text -->
```

必需项或可选项？

在后面各章节的元素定义中（通常由标题**格式**标识），除非元素属性及子元素指明为“（必需）”，否则均为可选项。要了解完整的元素语法信息，请参阅附录 A、CLEF XML 架构第 224 页码。

规范文件的示例

下面是一个完整的 CLEF 规范文件示例，本例适用于简单的数据转换节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0" debug="true">
  <ExtensionDetails id="urlparser" providerTag="spss" label="URL CLEF Module" version="1.0"
  provider="IBM Corp." copyright="(c) 2005-2011 IBM Corp." description="A Url Transform CLEF Extension"/>
  <Resources>
    <SharedLibrary id="urlparser_library" path="spss.urlparser/urlparser" />
  </Resources>
  <Node id="urlparser_node" type="dataTransformer" palette="fieldOp" label="URL Parser">
    <Properties>
      <Property name="id_fieldname" valueType="integer" label="ID field" />
      <Property name="url_fieldname" valueType="string" label="URL field" />
    </Properties>
    <UserInterface>
      <Icons />
      <Tabs>
        <Tab label="Types" labelKey="optionsTab.LABEL">
          <PropertiesPanel>
            <SingleFieldChooserControl property="id_fieldname" storage="integer" />
            <SingleFieldChooserControl property="url_fieldname" storage="string" />
          </PropertiesPanel>
        </Tab>
      </Tabs>
      <Controls />
    </UserInterface>
    <Execution>
      <Module libraryId="urlparser_library" name="">
        <StatusCodes>
          <StatusCode code="0" status="error" message="Cannot initialise a peer" />
          <StatusCode code="1" status="error" message="error reading input data" />
          <StatusCode code="2" status="error" message="Internal Error" />
          <StatusCode code="3" status="error" message="Input Field Does Not Exist" />
        </StatusCodes>
      </Module>
    </Execution>
    <OutputDataModel mode="replace">
      <AddField name="{id_fieldname}" fieldRef="{id_fieldname}" />
      <AddField name="{url_fieldname}" fieldRef="{url_fieldname}" />
      <AddField name="{url_fieldname}_server" storage="string" />
      <AddField name="{url_fieldname}_path" storage="string" />
      <AddField name="{url_fieldname}_field" storage="string" />
      <AddField name="{url_fieldname}_value" storage="string" />
    </OutputDataModel>
  </Node>
</Extension>
```

```

        </OutputDataModel>
    </Node>
</Extension>

```

ExtensionDetails 元素提供有关在 IBM® SPSS® Modeler 内部使用的扩展的基本信息。

Resources 元素用于指定服务器端库的位置，该服务器端库稍后将在文件中引用。路径规范表明该库位于 `\ext\bin\spss.urlparser\urlparser.dll` 下的 SPSS Modeler 安装目录中。

这个特定的规范文件并不包含 **CommonObjects** 元素。

Node 元素指定与节点本身有关的所有信息：

- 初始状态下，**Properties** 元素下具有两个已声明的属性，可供稍后在节点对话框的选项卡中使用。
- **UserInterface** 元素定义此扩展专用的节点对话框选项卡的外观和布局（其他选项卡由 SPSS Modeler 提供）。
- **Execution** 元素则定义执行节点时所使用的项。在本例中，这些项是之前在文件中声明的服务器端库，以及要在执行返回特定状态代码时显示的一组消息。
- **OutputDataModel** 元素定义此节点执行的数据转换。该元素指定输入数据模型（输入到此节点中的字段的集）将由此处定义的字段集替代，这些字段集将构成输出数据模型（从此处传递到所有下游节点的字段集，除非此后对模型进一步修改）。在这个特定的示例中，节点并未更改传递的两个原始字段（`id_fieldname` 和 `url_fieldname`），而是另外添加了四个字段，且其名称派生自 `url_fieldname`。

这个特定的规范文件是从作为 IBM® SPSS® Modeler 安装程序的一部分提供的示例节点之一中提取出来的。有关详细信息，请参阅第 31 页码第 3 章中的数据转换器节点（URL Parser）。

XML 声明

XML 声明是可选项，用于指定当前所用 XML 的版本，同时还提供字符编码格式的详细信息。

示例

```
<?xml version="1.0" encoding="UTF-8"?>
```

扩展元素

扩展元素是文件的主要组成部分，包含其他所有章节。格式为：

```

<Extension version="version_number" debug="true_false">
    Extension Details section
    Resources section
    Common Objects section
    User Interface (Palettes) section
    Object Definition section

```



```
</Extension>
```

其中：

`version` 是扩展的版本号。

`debug` 是可选项；如果设置为 `true`，将在与 CLEF 节点或输出相关的任意对话框或框中添加调试选项卡，同时还提供对该对象定义的属性和容器的访问。默认值是 `false`。
有关详细信息，请参阅第 220 页码第 10 章中的使用“调试”选项卡。

扩展详细信息部分

扩展详细信息部分提供与扩展相关的基本信息。

格式

```
<ExtensionDetails providerTag="extension_provider_tag"
  id="extension_unique_identifier"
  label="display_name" version="extension_version_number"
  provider="extension_provider" copyright="copyright_notice"
  description="extension_description"/>
```

其中：

`providerTag`（必需）是唯一标识此扩展的提供程序的名称。请注意，值不应包含字符串 `spss`，这是仅供内部使用的预留字符串。

`id`（必需）是唯一标识此扩展的名称，用于相关的系统消息中。按照惯例，扩展文件将放置在 IBM® SPSS® Modeler 安装目录中名为 `\ext\lib\providerTag.id` 的文件夹中。

`label`（必需）是扩展的显示标签。添加节点后，此文本将显示在选项板管理器的“名称”字段中。有关详细信息，请参阅第 219 页码第 10 章中的测试 CLEF 扩展。

`version` 是此扩展的版本号。

`provider` 是标识此扩展的提供程序的字符串。添加节点后，此文本将显示在选项板管理器的“提供程序”字段中。默认值为字符串 `(unknown)`。

`copyright` 是此扩展的版权声明。添加节点后，此文本将显示在选项板管理器的“版权”字段中。

`description` 是创建此扩展目的的简短说明。添加节点后，此文本将显示在选项板管理器的“说明”字段中。

示例

```
<ExtensionDetails providerTag="myco" id="sorter" name="Sort Data" version="1.2"
  provider="My Company Inc." copyright="(c) 2005-2006 My Company Inc."
  description="An example extension that sorts data using built-in OS commands."/>
```

资源部分

本节定义要想此扩展正常发挥作用需要哪些外部资源。

格式

```
<Resources>
  <Bundle .../>
  ...
  <JarFile .../>
  ...
  <SharedLibrary .../>
  ...
  <HelpInfo .../>
</Resources>
```

其中：

Bundle 标识一组位于客户端的已本地化资源。 [有关详细信息，请参阅第 38 页码数据包。](#)

JarFile 标识位于客户端的 Java jar 文件。 [有关详细信息，请参阅第 39 页码 Jar 文件。](#)

SharedLibrary 标识服务器端库或 DLL。 [有关详细信息，请参阅第 39 页码共享库。](#)

HelpInfo 指定扩展帮助信息的类型（如果有）。 [有关详细信息，请参阅第 182 页码第 7 章中的实施帮助系统。](#)

示例

```
<Resources>
  <SharedLibrary id="discriminantnode" path="spss.xd/Discriminant"/>
  <Bundle id="translations.discrim" type="properties" path="messages"/>
  <JarFile id="java" path="discriminant.jar"/>
  <HelpInfo id="help" type="native"/>
</Resources>
```

数据包

Bundle 元素指定位于客户端的资源数据包（例如一组用于本地化的消息文本），它可作为 `.properties` 文件或 Java `.class` 文件实施。 [有关详细信息，请参阅第 185 页码第 8 章中的本地化。](#)

格式

```
<Bundle id="identifier" path="path"/>
```

其中：

`id`（必需）是此数据包的唯一标识符。

`path`（必需）指定数据包文件相对于此规范文件父文件夹的位置。其中，数据包指 `.properties` 文件，该路径不能包含语言扩展或 `.properties` 后缀。

示例

```
<Bundle id="translations.discrim" path="messages"/>
```

这表示在规范文件所在的同一个文件夹中，一个名为 `messages.properties` 的文件中存在资源数据包。

Jar 文件

`JarFile` 元素指定客户端 Java 存档（`.jar`）文件，该文件为此扩展提供 Java 类和其他客户端资源。

格式

```
<JarFile id="identifier" path="path"/>
```

其中：

`id`（必需）是此 `.jar` 文件的唯一标识符。

`path`（必需）指定 `.jar` 文件相对于此规范文件父文件夹的位置。

示例

```
<JarFile id="java" path="coxreg_model_terms.jar"/>
```

这表示此扩展的 `.jar` 文件位于规范文件所在的文件夹中。

共享库

`SharedLibrary` 元素指定服务器端共享库或 DLL。此元素通常仅在支持节点执行时才需要使用。在一个实施了多个模块的库中，节点规范执行部分中的 `Module` 元素用于标识库中的一个具体模块。

格式

```
<SharedLibrary id="identifier" path="path"/>
```

其中：

`id`（必需）是此共享库的唯一标识符。

`path`（必需）指定共享库相对于服务器端安装目录中的 `\ext\bin` 文件夹的位置。请注意，该路径不能包含共享库文件扩展名（例如 `.dll`）。

示例

下面的共享库声明：

```
<SharedLibrary id="binning" path="spss.binning/Binning" />
```

指定共享库加载自：

```
install_dir\ext\bin\spss.binning\Binning.dll
```

其中，install_dir 是服务器端 CLEF 组件的安装目录。由于此库实施了多个模块，所需的特定模块（supervisedBinning）由构建节点规范中的 Module 元素标识，此时将引用如下库标识符：

```
<Execution>  
  <Module libraryId="binning" name="supervisedBinning" .../>  
  ...  
</Execution>
```

帮助信息

可选的 HelpInfo 元素指示能为此扩展提供哪些类型的帮助。 [有关详细信息，请参阅第 182 页码第 7 章中的实施帮助系统。](#)

通用对象部分

可选的通用对象部分用于定义可以在规范文件其他位置定义的元素间共享的对象。此部分中某些类型的对象（例如属性枚举）也可以根据需要在本地定义，但其他对象（例如模型和文档）只能在此定义。

格式

```
<CommonObjects>  
  <PropertyTypes .../>  
  <PropertySets .../>  
  <ContainerTypes .../>  
  <Actions .../>  
  <Catalogs .../>  
</CommonObjects>
```

其中：

PropertyTypes 使得通用属性定义可以在对象间共享。 [有关详细信息，请参阅第 41 页码属性类型。](#)

PropertySets 通常会在模型构建器节点、模型输出对象和模型填充器节点包含相同属性集时使用。 [有关详细信息，请参阅第 42 页码属性集。](#)

ContainerTypes 用于定义容器的类型，容器就是可容纳复杂数据结构的对象。 [有关详细信息，请参阅第 43 页码容器类型。](#)

Actions用于定义有关用户交互的基本信息，例如通过菜单或工具栏进行交互。有关详细信息，请参阅第 44 页码Actions。

Catalogs 实现可从由服务器动态生成的值列表选择一个或多个选项的控件。有关详细信息，请参阅第 46 页码Catalogs。

示例

```
<CommonObjects>
  <ContainerTypes>
    <ModelType id="discriminant_model" format="utf8" />
    <DocumentType id="html_output" />
    <DocumentType id="zip_outputType" format="binary"/>
  </ContainerTypes>
</CommonObjects>
```

属性类型

可选的属性类型部分使得通用的属性定义可以在对象间共享。这在某种程度上可以减轻维护工作量，例如，某个属性的定义可以出现在一个位置，而不是在多个位置重复出现。定义共享也可用于确保当创建新的对象实例时，不同对象中复制了值的属性间相互兼容。

属性类型只能在通用对象部分定义。

格式

```
<PropertyTypes>
  <PropertyType id="identifier" isKeyed="true_false" isList="true_false" max="max_value"
    min="min_value" valueType="value_type">
    <Enumeration .../>
    <Structure .../>
    <DefaultValue .../>
  </PropertyType>
  <PropertyType .../>
  ...
</PropertyTypes>
```

PropertyType 属性如下所示。

id（必需）是属性类型的唯一标识符。

isKeyed，如果设置为 **true**，则表示这是键控类型的属性。键控属性通过用户自定义控件将一组操作关联到某个字段（请参阅属性控件第 153 页码）。如果 **isKeyed** 设置为 **true**，**valueType** 属性必须设置为 **structure**。有关结构化属性的详细信息，请参阅结构化属性第 69 页码。

isList 指定该属性是特定值类型的一列值 (**true**) 还是单个值 (**false**)。

max 和 **min** 表示值范围的最大值和最小值。

`valueType` 可以是以下类型之一：

- `string`
- `encryptedString`
- `fieldName`
- `integer`
- `double`
- `boolean`
- `date`
- `enum`（请参阅 枚举属性第 68 页码）
- `structure`（请参阅 结构化属性第 69 页码）
- `databaseConnection`

`Enumeration` 和 `Structure` 子元素相互排斥。在特定的情况下，还会使用 `Enumeration`、`Structure` 和 `DefaultValue` 子元素，请参阅 枚举属性第 68 页码、结构化属性第 69 页码 和 默认值第 71 页码。

属性集

当模型构建器节点、模型输出对象和模型填充器节点包含同一组属性时，通常会用到属性集。例如，模型构建器节点可以定义一个在模型应用前已在构建器中设定但并不实际使用的默认属性集。为了能够自动传输属性集，还必须将它们包含在模型输出中。

格式

```
<PropertySets>
  <PropertySet id="identifier">
    <Property .../>
    <Property .../>
    ...
  </PropertySet>
  ...
</PropertySets>
```

其中，`id` 是此属性集的唯一标识符。

有关 `Property` 元素的说明，请参阅[Properties](#)第 58 页码。

示例

本示例演示的定义由两个属性组成：要生成的预测数，以及是否包含概率。在通用对象部分，您将定义：

```
<PropertySets>
  <PropertySet id="common_model_properties">
    <Property name="prediction_count" valueType="integer" min="1" max="10"/>
    <Property name="include_probabilities" valueType="boolean" defaultValue="false"/>
  </PropertySet>
</PropertySets>
```

```

    </PropertySet>
    ...
</PropertySets>

```

然后，在模型构建器节点、模型输出对象和模型填充器节点各自的定义中，都将包含一个 `includePropertySets` 属性，如下所示（此处仅举例说明模型构建器节点的定义）：

```

<Node id="my_builder" type="modelBuilder" ...>
  <Properties includePropertySets="[common_model_properties]">
    ...
  </Properties>
  ...
</Node>

```

容器类型

容器就是作为复杂数据结构（如模型和文档）的占位符使用的对象。容器定义总是包含具体的类型，而容器类型将在此处定义。可以定义下列容器类型：

- 模型类型
- 文档类型

容器类型可以在客户端和服务器之间传输，克隆后再保存到文件或内容存储库中。当从模型输出对象生成模型填充器节点时，将会对模型进行克隆。

每种类型的容器都具有一个预定义的属性集，不过，可以添加自定义属性。容器类型只能在通用对象部分定义。

格式

容器类型部分的格式为：

```

<ContainerTypes>
  <ModelType .../>
  ...
  <DocumentType .../>
  ...
</ContainerTypes>

```

其中：

ModelType 指定特定模型类型的格式。 [有关详细信息，请参阅第 44 页码模型类型。](#)

DocumentType 指定特定文档类型的格式。 [有关详细信息，请参阅第 44 页码文档类型。](#)

示例

```

<ContainerTypes>
  <ModelType id="discriminant_model" format="utf8">
  <DocumentType id="html_output" />

```

```
<DocumentType id="zip_outputType" format="binary"/>
</ContainerTypes>
```

模型类型

模型必须提供算法名称、模型类型以及输入和输出数据模型等信息。模型类型定义用于指定特定模型类型的格式。

模型类型信息可以在规范文件中的这一部分静态指定，或者也可以在模型构建器节点构建模型时动态指定。

格式

```
<ModelType id="identifier" format="model_type_format" />
```

其中：

- `id`（必需）是模型类型的唯一标识符。
- `format`（必需）是模型类型的格式，可以是 `utf8`（文本），也可以是 `binary`。必须将模型格式指定为静态信息的一部分。

示例

```
<ModelType id="my_model" format="utf8" />
```

文档类型

`document` 是一种输出对象，如图形或报告。文档类型定义用于指定特定文档类型的格式。

格式

```
<DocumentType id="identifier" format="document_type_format" />
```

其中：

- `id`（必需）是文档类型的唯一标识符。
- `format`（必需）是文档类型的格式，可以是 `utf8`（文本），也可以是 `binary`。

示例

```
<DocumentType id="html_output" format="utf8" />
<DocumentType id="zip_outputType" format="binary"/>
```

Actions

操作用于定义有关用户交互的基本信息，例如通过菜单或工具栏进行交互。每个操作定义标签、工具提示或图标等在用户界面中的表示方式。一系列的操作则由为各组操作定义的客户端 `Java` 类处理。操作也可以在具体的对象中定义。

格式

```
<Actions>
  <Action id="identifier" label="display_label" labelKey="label_key" description="action_description"
    descriptionKey="description_key" imagePath="image_path" imagePathKey="image_path_key"
    mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" shortcut="shortcut_string"
    shortcutKey="shortcut_key" />
  ...
</Actions>
```

其中：

id（必需）是操作的唯一标识符。

label（必需）是操作显示在用户界面上时的显示名称。

labelKey 用于标识标签以便进行本地化。

description 是操作的说明，例如，对于自定义菜单项或工具栏上的图标操作按钮，这将是该菜单项或按钮的工具提示文本。

descriptionKey 用于标识说明以便进行本地化。

imagePath 是图形文件（如图标图像）的位置。提供的位置相对于规范文件的安装目录。

imagePathKey 用于标识本地化时使用的图像路径。

mnemonic 是与 Alt 键一起使用以激活此控件的字母符号（例如，如果您给其值为 S，则用户可通过 Alt-S 激活此控件）。

mnemonicKey 标识助记符以便进行本地化。如果不使用 **mnemonic** 和 **mnemonicKey**，则此控件无可用的助记符。有关详细信息，请参阅第 126 页码第 6 章中的[访问键和键盘快捷键](#)。

shortcut 是指示可用于启动操作的键盘快捷键（例如，CTRL+SHIFT+A）的字符串。

shortcutKey 用于标识快捷键以便进行本地化。如果不使用 **shortcut** 和 **shortcutKey**，则该操作无可用的快捷键。有关详细信息，请参阅第 126 页码第 6 章中的[访问键和键盘快捷键](#)。

示例

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP"/>
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Generates a derive node"
    descriptionKey="generate.deriveNode.TOOLTIP"/>
</Actions>
```

Catalogs

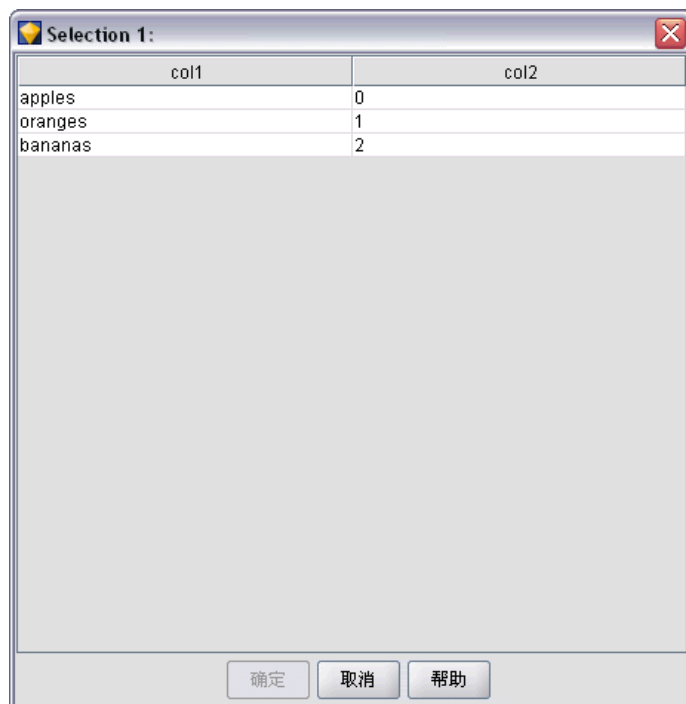
目录可以将某个属性与控件关联，以允许用户从由服务器动态生成的值列表中选择一个或多个选项。

图片 4-1
允许从值列表进行选择的控件



当用户单击 <Select> 条目时，值以弹出列表形式显示在控件中。

图片 4-2
显示的值列表



当用户在列表中选择一行时，在 `Catalog` 元素中的指定列上的行值将被放入控件中。

图片 4-3
从值列表进行选择



格式

```
<CommonObjects>
  <Catalogs>
    <Catalog id="identifier" valueColumn="integer">
      <Attribute label="display_name" />
      ...
    </Catalog>
    ...
  </Catalogs>
</CommonObjects>
```

```
</Catalogs>
</CommonObjects>
```

其中：

id（必需）是目录的唯一标识符。

valueColumn（必需）是当用户选择某行时其值将被放入控件中的列的编号。列从 1 开始编号。

按列为每个列使用一个 **Attribute** 元素，请参阅下面的示例。

当用户激活与目录关联的控件时，通过调用 `getCatalogInformation` 函数从服务器检索到包含值列表的目录。此函数返回包含值列表的 XML 文档。[有关详细信息，请参阅第 196 页码第 9 章中的对等函数。](#)

示例

本示例演示之前显示的定义目录控件所使用的部分代码。这里定义了三个目录，并将其关联到对话框选项卡上的三个不同控件。

首先，在通用对象部分中定义目录：

```
<CommonObjects>
  <Catalogs>
    <Catalog id="cat1" valueColumn="1">
      <Attribute label="col1" />
      <Attribute label="col2" />
    </Catalog>
    <Catalog id="cat2" valueColumn="2">
      <Attribute label="col1" />
      <Attribute label="col2" />
      <Attribute label="col3" />
    </Catalog>
    <Catalog id="cat3" valueColumn="1">
      <Attribute label="col1" />
    </Catalog>
  </Catalogs>
</CommonObjects>
```

然后，在节点定义的“属性”部分定义与控件关联的属性：

```
<Node id="catalognode" type="dataReader" palette="import" label="Catalog">
  <Properties>
    <Property name="sometext" valueType="string" label="Some Text" />
    <Property name="selection1" valueType="string" label="Selection 1" />
    <Property name="selection2" valueType="string" isList="true" label="Selection 2" />
    <Property name="selection3" valueType="string" label="Selection 3" />
  </Properties>
```

在节点定义的“用户界面”部分中，对控件进行定义并通过属性引用将其关联到目录定义：

```

<User Interface>
  <Tabs>
    <Tab label="Catalog Controls" labelKey="Catalog.LABEL" >
      <PropertiesPanel>
        <TextBoxControl property="sometext" />
        <SingleItemChooserControl property="selection1" catalog="cat1" />
          <MultiItemChooserControl property="selection2" catalog="cat2" />
          <SingleItemChooserControl property="selection3" catalog="cat3" />
        </PropertiesPanel>
      </Tab>
    </Tabs>
  </User Interface>

```

用户界面（选项板）部分

这是一个可选部分，且仅当要将此扩展用于定义在其上显示节点的自定义选项板或子选项板时才会包括这一部分。

在定义了自定义选项板或子选项板的扩展中，当后续加载的扩展用于定义要包括在同一个选项板或子选项板中的节点时，可以省略此用户界面（选项板）部分，此时只需 `Node` 元素包含引用此选项板的 `customPalette` 属性即可。扩展按照 `providerTag.id` 值的字母顺序加载，其中，这些是此扩展的 `ExtensionDetails` 元素的 `providerTag` 和 `id` 属性的值（请参阅[扩展详细信息部分第 37 页码](#)）。因此，将先加载扩展 `myco.abc`，再加载扩展 `myco.def`。

注意：用户界面（选项板）部分与主用户界面部分不同，前者显示为单个对象定义的一部分，有关详细信息，请参阅第 6 章，构建用户界面 第 113 页码。

格式

用户界面（选项板）部分的格式为：

```

<User Interface>
  <Palettes>
    <Palette id="name" systemPalette="palette_name" customPalette="palette_name"
      relativePosition="位置" relativeTo="palette" label="display_label"
      labelKey="label_key" description="description" descriptionKey="description_key"
      imagePath="image_path" />
    <Palette .../>
    ...
  </Palettes>
</User Interface>

```

表 4-1
选项板属性

特性	描述
id	(必需) 要定义的选项板或子选项板的唯一标识符。
systemPalette	<p>仅当在系统选项板中添加子选项板时使用，用于标识将显示此子选项板的系统选项板：</p> <ul style="list-style-type: none"> import - 源 recordOp - 记录选项 fieldOp - 字段选项 graph - 图形 modeling - 建模 (请参阅下文) dbModeling - 数据库建模 output - 输出 export - 导出
customPalette	<p>仅当在自定义选项板中添加子选项板时使用，用于标识将显示此子选项板的自定义选项板。这是定义自定义选项板的 Palette 元素的 id 属性的值。</p>
relativePosition	<p>仅当定义自定义选项板时使用，用于指定选项板在屏幕底部的选项板栏中的位置。</p> <p>可能的值包括：</p> <ul style="list-style-type: none"> first last before after <p>如果值为 before 或 after，则还要求具有 relativeTo 属性 (请参阅下文)。</p> <p>如果省略 relativePosition，此选项板将位于选项板栏的最后。</p>
relativeTo	<p>如果 relativePosition 的值为 before 或 after，则 relativeTo 用于指定此自定义选项板之前或之后的选项板的标识符。选项板标识符将作为 Node 元素的选项板属性值列出 (请参阅 节点第 54 页码)。</p>
label	(必需) 当选项板或子选项板显示在用户界面上时所显示的名称。
labelKey	用于标识标签以便进行本地化。
description	<p>当鼠标悬停在选项板选项卡 (不适用于子选项板) 上时显示的工具提示文本。此值还可作为控件的详细辅助说明。 有关详细信息，请参阅第 191 页码第 8 章中的辅助功能。</p>
descriptionKey	用于标识说明以便进行本地化。
imagePath	<p>用于标识所用图像在选项板选项卡 (不适用于子选项板) 上的位置。该位置根据规范文件安装所在的目录进行指定。如果您省略此属性，将不使用任何图像。</p>

示例 - 添加节点至系统选项板

假设您的组织发明了一种新的挖掘音频和视频数据的算法，您要将此算法集成到 IBM® SPSS® Modeler 中。您从定义自定义数据读取器节点开始，该节点将用于从音频和视频文件中读取输入数据。

开始时，您决定将新的数据读取器节点添加到“源”系统选项板中。您只需通过 **Node** 元素的 **palette** 属性来标识源选项板。有关详细信息，请参阅第 54 页码节点。

使用此方法将此节点添加到“源”选项板中的“数据库”节点之后，可以使用：

```
<Node id="AVreader" type="dataReader" palette="import" relativePosition="after" relativeTo="database"
  label="AV Reader">
```

结果是：

图片 4-4
标准选项板上的新节点



示例 - 添加自定义选项板

虽然使用标准 IBM® SPSS® Modeler 选项板就能够满足要求，但您希望新节点显得更加突出。所以您决定为此节点定义一个自定义选项板，并且将该选项板置于“收藏夹”选项板之后、“源”选项板之前。首先，您需要添加用户界面（选项板）部分，以便按照下列操作定义自定义选项板：

```
<User Interface>
  <Palettes>
    <Palette id="AV_mining" label="AV Mining" relativePosition="before" relativeTo="import"
      description="Audio video mining" />
  </Palettes>
</User Interface>
```

relativeTo 属性必须使用“源”选项板的内部标识符，即 **import**。

然后按照下列操作更改 **Node** 定义：

```
<Node id="AVreader" type="dataReader" customPalette="AV_mining" label="AV Reader">
```

这样会将 AV 挖掘选项板置于“收藏夹”和“源”系统选项板之间。

图片 4-5
新选项板上的新节点



示例 - 添加自定义子选项板至自定义选项板

继续使用上一个示例，您现在更希望将数据读取器节点置于 AV 挖掘选项板上自己的 AV 源子选项板中。为此，您首先需要通过在用户界面（选项板）部分中添加另一个 Palette 元素来指定子选项板：

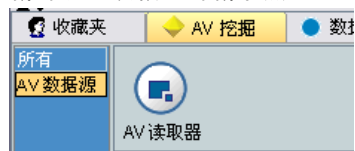
```
<UserInterface>
  <Palettes>
    <Palette id="AV_mining" label="AV Mining" description="Audio video mining" />
    <Palette id="AV_mining.sources" customPalette="AV_mining" label="AV Sources" />
  </Palettes>
</UserInterface>
```

然后，更改 Node 元素，以引用子选项板的标识符：

```
<Node id="AVreader" type="dataReader" customPalette="AV_mining.sources" label="AV Reader">
```

现在，当用户单击 AV 挖掘选项卡时，他们将看到两个子选项板，一个标记为所有，而另一个标记为 AV 源。这两个子选项板上均显示“AV 读取器”节点：

图片 4-6
新的子选项板上的新节点



如果您在 AV 挖掘的另一个新子选项板中再添加一个新的节点，这个新节点将同时显示在所有和新的子选项板中，但不会显示在 AV 源子选项板中。

示例 - 添加节点至系统子选项板

为了处理音频和视频源数据，您现在定义了一个模型构建器节点。您决定将这个节点添加到标准的“建模”选项板中，该选项板包含多个标准子选项板。您选择将其添加到“分类”子选项板中，并置于“神经网络”节点之前，因此需要指定：

```
<Node id="AVmodeler" type="modelBuilder" palette="modeling.classification" relativePosition="before"
  relativeTo="neuralnet" label="AV Modeler">
```

现在，“分类”子选项板将显示为：

图片 4-7
标准子选项板上的新节点



请注意，还将在“建模”选项板的“所有”子选项板上的同一个相对位置添加该节点。

示例 - 添加自定义子选项板至系统选项板

再次查看“分类”子选项板中模型构建器节点的数目，您意识到用户可能不容易注意到您添加的新节点。要让您的节点显得更为突出，一种方法就是在“建模”选项板中添加自己的子选项板并在其中放置该节点。

因此，您必须先通过在文件中添加用户界面（选项板）部分来定义自定义子选项板：

```
<User Interface>
  <Palettes>
    <Palette id="modeling.av_modeling" systemPalette="modeling" label="AV Modeling"
      labelKey="av_modeling.LABEL" description="Contains AV mining-related modeling nodes"
      descriptionKey="av_modeling.TOOLTIP"/>
  </Palettes>
</User Interface>
```

请注意，您必须明确指定将 `systemPalette` 用于标识要扩展的系统选项板。

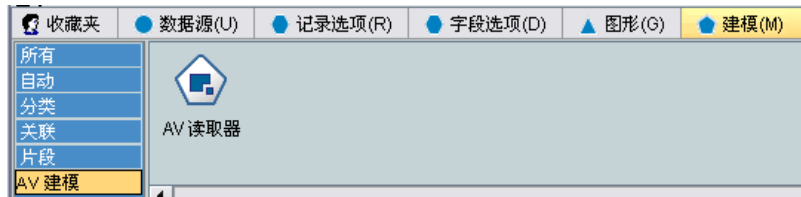
然后，在节点的主用户界面部分，您指定它在此子选项板中显示：

```
<Node id="my.avmodeler" type="modelBuilder" customPalette="modeling.av_modeling" label="AV
Modeler">
```

自定义子选项板始终位于系统子选项板之后，因此现在的结果为：

图片 4-8

添加到标准子选项板中的自定义新节点



注意：如果要在“AV 建模”子选项板中添加更多节点，只要先加载了 AV Modeler 扩展，这些节点的规范文件就不再需要用户界面（选项板）部分。

隐藏或删除自定义选项板或子选项板

如果不希望显示某个自定义选项板或子选项板，可以通过 IBM® SPSS® Modeler 选项板管理器隐藏或删除它。

请注意，隐藏操作会在多个 SPSS Modeler 会话中持续发挥作用，但此操作由复选框控制，因此可以恢复为正常显示。删除操作无法在同一会话中恢复，但重新启动 SPSS Modeler 后，被删除的项目会重新显示，除非您已从规范文件中将其删除，或者删除了整个扩展。有关详细信息，请参阅第 223 页码第 10 章中的卸载 CLEF 扩展。

要隐藏或删除选项板，请执行下列操作：

- ▶ 在 SPSS Modeler 主菜单中，选择：
工具 > 管理选项板

- ▶ 在“选项板名称”字段中选择所需选项板，然后：
 - 要隐藏选项板，请取消选中相应的“显示?”复选框
 - 要删除选项板，请单击“删除”选择按钮
 - ▶ 单击“确定”。
- 要隐藏或删除子选项板，请执行下列操作：
- ▶ 在 SPSS Modeler 主菜单中，选择：
工具 > 管理选项板
 - ▶ 在“选项板名称”字段中选择一个选项板。
 - ▶ 单击“子选项板”按钮。
 - ▶ 在“子选项板名称”字段中选择所需子选项板，然后：
 - 要隐藏子选项板，请取消选中相应的“显示?”复选框
 - 要删除子选项板，请单击“删除”选择按钮
 - ▶ 单击“确定”。

对象定义部分

扩展中的主要部分是元素。对象定义部分组成了 CLEF 规范文件的其余部分，用于定义扩展中的各种对象。可定义的对象类型包括：

- 节点
- 模型输出对象
- 文档输出对象
- 交互输出对象

节点就是显示在流中的对象。**模型输出对象**由模型构建器节点生成，显示在主窗口管理器窗格的“模型”选项卡下。类似地，**文档输出对象**由文档构建器节点生成，显示在同一个窗格的“输出”选项卡下。**交互输出对象**由交互模型构建器节点生成，显示在管理器窗格的“输出”选项卡下。

对象定义部分由一个或多个此类对象定义组成。

可以为不同类型对象定义的元素将在后面的部分介绍。这些元素中有一部分可通用于所有对象类型，而其他元素则特定于节点或模型输出定义。对象特定的元素在文本中由以下项目表示。

- 对象标识符
- 模型构建器
- 文档构建器
- 属性
- 容器
- 用户界面

- 执行
- 输出数据模型
- 构造函数

对象标识符

对象标识符用于标识对象的类型，包含以下几种：

```
<Node .../>
```

```
<ModelOutput .../>
```

```
<DocumentOutput .../>
```

```
<InteractiveModelBuilder .../>
```

对象标识符还提供有关对象如何通过脚本编写方式显示的信息。`scriptName` 属性表示对象的唯一名称。脚本可以使用此属性指定特定的对象（例如，流中的一个节点，或“输出”选项卡中的输出）。

节点

节点定义描述可在流中出现的对象。

格式

```
<Node id="identifier" type="node_type" palette="palette" customPalette="custom_palette"
  relativePosition="位置" relativeTo="节点" label="display_label" labelKey="label_key"
  scriptName="script_name" helpLink="topic_id" description="description"
  descriptionKey="description_key">
  <ModelBuilder ...>
  ...
  </ModelBuilder>
  <DocumentBuilder ...>
  ...
  </DocumentBuilder>
  <ModelProvider .../>
  <Properties>
  ...
</Properties>
  <Containers>
  ...
</Containers>
  <User Interface>
  ...
</User Interface>
  <Execution>
  ...
</Execution>
  <OutputDataModel ...>
```

```

...
  </OutputDataModel>
  <Constructors>
    ...
  </Constructors>
</Node>

```

允许在节点定义中使用的元素将在从[Properties](#)第 58 页码开始的章节中介绍。

表 4-2
节点属性

特性	描述
id	(必需) 该节点的标识符, 采用文本字符串格式。
type	<p>(必需) 节点的类型包括:</p> <p>dataReader - 用于读取数据的节点 (例如, “源” 选项板节点) dataWriter - 用于记录数据的节点 (例如, “导出” 选项板节点) dataTransformer - 用于转换数据的节点 (例如, “记录” / “字段选项” 节点) modelBuilder - 模型构建器节点 (例如, “建模” 选项板节点) documentBuilder - 用于创建图形或报告的节点 modelApplier - 包含生成的模型的节点</p> <p>节点类型决定着选项板和工作区中节点图标的形状。有关详细信息, 请参阅第 9 页码第 2 章中的节点概述。</p> <p>如果节点类型为 modelBuilder, 节点定义必须包含一个 ModelBuilder 元素, 请参阅模型构建器第 57 页码。</p> <p>如果节点类型为 documentBuilder, 节点定义必须包含一个 DocumentBuilder 元素, 请参阅文档构建器第 57 页码。</p>
palette	<p>要在其上显示节点的标准 IBM® SPSS® Modeler 选项板或子选项板之一的标识符, 名为:</p> <p>import - 源 recordOp - 记录选项 fieldOp - 字段选项 graph - 图形 modeling - 建模 (请参阅下文) dbModeling - 数据库建模 output - 输出 export - 导出</p> <p>“建模” 选项板包含多个标准子选项板:</p> <p>modeling.classification - 分类 modeling.association - 关联 modeling.segmentation - 细分 modeling.auto - 自动</p> <p>如果省略 palette 属性, 节点会显示在 “字段选项” 选项板中。</p> <p>注意: palette 属性仅用于模型构建器节点。</p>

特性	描述
customPalette	要在其上显示节点的自定义选项板或子选项板的标识符。这是 Palette 元素 id 属性的值，该值在文件的用户界面（选项板）部分指定。 有关详细信息，请参阅第 48 页码用户界面（选项板）部分。
relativePosition	指定节点在选项板中的位置。可能的值包括： first last before after 如果值为 before 或 after ，则还要求具有 relativeTo 属性（请参阅下文）。 如果省略 relativePosition ，此节点将位于选项板的最后。
relativeTo	如果 relativePosition 的值为 before 或 after ，则 relativeTo 用于指定此节点之前或之后的选项板的节点。 relativeTo 的值是节点的脚本名称。 对于标准 SPSS Modeler 节点，脚本名称可以在《SPSS Modeler脚本编写与自动化指南》的“属性参考”部分找到，但不带 ...node 后缀（例如，对于“数据库”节点，需要使用 database ，而不是 databasenode ）。 对于 CLEF 节点，此为该节点 scriptName 属性的值。
label	（必需）当节点出现在选项板、工作区和对话框中时显示的节点名称。
labelKey	用于标识标签以便进行本地化。
scriptName	用于在脚本中引用节点时唯一地标识节点。 有关详细信息，请参阅第 84 页码在脚本中使用 CLEF 节点。
helpLink	可选的帮助主题标识符，如果有，将在用户调用帮助系统时显示。标识符的格式取决于帮助系统的类型（请参阅第 7 章，添加帮助系统第 182 页码）： HTML Help - 帮助主题的 URL JavaHelp - 主题 ID
description	节点的文本说明。
descriptionKey	用于标识说明以便进行本地化。

可以包含在节点定义中的元素将在从[模型构建器第 57 页码](#)开始的章节中介绍。

示例

要查看节点定义的示例，请参阅[规范文件的示例第 35 页码](#)。

模型输出

模型输出定义描述生成的模型，即执行流后显示在管理器窗格中“模型”选项卡下的对象。

有关如何对文件的这一部分进行编码的详细信息，请参阅[模型输出第 95 页码](#)。

文档输出

文档输出定义描述生成的表或图形等对象，此类对象将在执行流后显示在管理器窗格中的“输出”选项卡下。

有关如何对文件的这一部分进行编码的详细信息，请参阅[文档输出第 108 页码](#)。

交互模型构建器

有关如何对文件的这一部分进行编码的详细信息，请参阅[构建交互模型第 96 页码](#)。

模型构建器

此元素仅在 Node 元素定义中使用。

有关如何对文件的这一部分进行编码的详细信息，请参阅[第 5 章，构建模型和文档第 86 页码](#)。

文档构建器

此元素仅在 Node 元素定义中使用。

有关如何对文件的这一部分进行编码的详细信息，请参阅[第 5 章，构建模型和文档第 86 页码](#)。

模型提供器

此元素仅在 Node 元素定义中使用。

当定义模型输出对象和模型填充器节点时，您可以使用 **ModelProvider** 元素指定用于容纳该模型的容器。您也可以指定是否要以 PMML 格式存储该模型。PMML 模型可以通过自定义查看器查看，也可以通过标准 IBM® SPSS® Modeler 模型输出查看器查看，该标准查看器由 **ModelViewerPanel** 元素提供。[有关详细信息，请参阅第 133 页码第 6 章中的模型查看器面板](#)。

格式

```
<ModelProvider container="container_name" isPMML="true_false" />
```

其中：

container 是用于容纳模型的容器的名称。

isPMML 指示是否以 PMML 格式存储模型。

示例

```
<ModelProvider container="model" isPMML="true" />
```

要查看在模型填充器节点的上下文中使用 `ModelProvider` 的示例，请参阅[模型查看器面板第 133 页码](#)下的示例。

Properties

属性定义包含一组由名称和值组成的数据对。单个属性定义（可能会有多个）分别包含在单独的属性部分中。

注意：如果已在属性部分中定义某个属性，则无需为单个属性控件定义该属性，因为属性部分定义的优先级较高。因此，我们建议在属性部分定义属性。

此规则的一个例外是与 `label` 属性相关的情况。如果已经为属性控件定义了 `label` 属性，则在属性控件声明中发现的任意属性定义（不仅仅是 `label` 定义）比属性部分中的相应定义具有较高的优先级。请注意，此例外情况仅适用于属性控件，而不适用于其他类型的控件，如菜单、菜单项和工具栏项目。这些项目必须明确定义一个标签，可通过 `Action` 元素直接定义（菜单）或间接定义（菜单项和工具栏项目）。

格式

```
<Properties>
  <Property name="name" scriptName="script_name" valueType="value_type" isList="true_false"
    defaultValue="default_value" label="display_label" labelKey="label_key" description="description"
    descriptionKey="description_key" />
  <Enumeration .../>
  <Structure .../>
  <DefaultValue .../>
  ...
</Properties>
```

在特定的情况下，还会使用 `Enumeration`、`Structure` 和 `DefaultValue` 元素。[有关详细信息，请参阅第 67 页码值类型。](#)

`Property` 元素的属性如下所示：

表 4-3
Property 属性

特性	描述
<code>name</code>	（必需）属性的唯一名称。
<code>scriptName</code>	属性在脚本中所引用的名称。 有关详细信息，请参阅第 84 页码在脚本中使用 CLEF 节点。

特性	描述
valueType	指定此属性可以采用的值类型，包括： string encryptedString fieldName integer double boolean date enum structure databaseConnection 有关详细信息，请参阅第 67 页码值类型。
isList	指定该属性是特定值类型的一列值 (<code>true</code>) 还是一个值 (<code>false</code>)。
defaultValue	此属性的默认值。这可以表示为一个简单的值属性或一个复杂的元素，而且必须与指定的有效值一致。
label	当属性值显示在用户界面上时的显示名称。
labelKey	用于标识标签以便进行本地化。
description	属性的说明。
descriptionKey	用于标识说明以便进行本地化。

属性可以选择声明如何确定有效值：

- 对于数字值，这将是 最小和/或最大值。
- 对于字符串，这通常是 字段选择（例如，所有字段、所有数字字段、所有离散字段等）但也可能是 文件选择。
- 对于枚举，这将是 有效值的集合。

键控属性也必须声明如何确定有效键。请注意，键控属性的键类型必须是字符串或枚举。 [有关详细信息，请参阅第 41 页码属性类型。](#)

创建属性的相关对象之后，将对与该属性相关的可选默认值进行求值。例如，每次创建新节点实例时，都会对节点属性的默认值进行求值；每次执行节点时，都会对执行属性进行求值。求值按照属性的声明顺序执行。

请注意，属性定义可以引用通用对象部分中声明的属性类型。

Containers

容器就是在构造函数部分定义其生成的输出对象的占位符。

格式

```
<Containers>
  <Container name="container_name" />
  ...
</Containers>
```

其中：

`name` 相当于 `CreateModel` 或 `CreateDocument` 元素的目标属性值（请参阅[使用构造函数第 109 页码](#)），而且间接地将容器与在通用对象部分中声明的一种容器类型关联起来。

示例

首先，在通用对象部分中声明容器类型。为模型声明了一种容器类型，即采用文本格式；为文档输出对象声明了两种容器类型，一种是对 HTML 输出采用的默认（文本）格式，一种是对压缩的输出采用的二值格式。

```
<CommonObjects>
  <ContainerTypes>
    <ModelType id="my_model" format="utf8" />
    <DocumentType id="html_output" />
      <DocumentType id="zip_outputType" format="binary" />
    </ContainerTypes>
  </CommonObjects>
```

在节点定义的执行部分，将输出文件定义为包含容器类型（对应于通用对象部分中指定的标识符）的容器文件：

```
<Node id="mynode" ...>
  ...
  <Execution>
    ...
  <OutputFiles>
    <ContainerFile id="pmm1" path="{tempfile}.pmm1" containerType="my_model" />
      <ContainerFile id="htmloutput" path="{tempfile}.html" containerType="html_output" />
      <ContainerFile id="zipoutput" path="{tempfile}.zip" containerType="zip_outputType" />
    </OutputFiles>
```

此后，构造函数部分定义执行节点时要生成的输出对象。此处，`CreateModel` 和 `CreateDocument` 元素都具有与容器文件对应的 `sourceFile` 属性，正如之前在“输出文件”部分中指定的一样：

```
  <Constructors>
    <CreateModelOutput type="myoutput">
      <CreateModel target="模型" sourceFile="pmm1" />
        <CreateDocument target="advanced_output" sourceFile="htmloutput" />
        <CreateDocument target="zip_output" sourceFile="zipoutput" />
      </CreateModelOutput>
    </Constructors>
  </Execution>
</Node>
```

最后，模型输出部分将容器与模型输出对象或文档输出对象关联。在 `Container` 元素中，`name` 属性对应于指定的 `CreateModel` 和 `CreateDocument` 元素中的 `target` 属性：

```
<ModelOutput id="myoutput" label="My Model">
  <Containers>
    <Container name="模型" />
```



```

        <Container name="advanced_output" />
        <Container name="zip_output" />
    </Containers>
    ...
</ModelOutput>

```

用户界面

规范文件支持多种用户界面组件，以便显示各种对象和修改各种控件和属性。有多种工具可供用户使用，用于指定组件的布局和调整大小行为，以及如果修改了其他控件之后，是否应该启用组件或显示组件。

用户界面部分用于指定对象的可视外观。规范可用于自定义基本的用户界面组件，例如节点属性对话框或输出窗口。

用户界面部分是 **Node** 元素规范的必需部分。

有关如何对文件的这一部分进行编码的详细信息，请参阅 [第 6 章，构建用户界面](#) [第 113 页码](#)。

执行

此元素仅在 **Node** 元素定义中使用。

执行部分用于定义执行节点时使用的属性和文件。

格式

```

<Execution>
  <Properties>
    ...
  </Properties>
  <InputFiles>
  <ContainerFile .../>
    ...
  </InputFiles>
  <OutputFiles>
    <ContainerFile .../>
    ...
  </OutputFiles>
  <Module ...>
    <StatusCodes .../>
  </Module>
  <Constructors .../>
</Execution>

```

执行部分包括每次执行节点时都会重新创建的一组属性的定义，这些属性仅在执行节点时可用。

执行信息也可以定义要在执行节点之前生成的输入文件集，以及在执行时生成的任意输出文件。

可以指定任意数量的输入文件和输出文件。每个输入文件都具有一个由节点定义的关联容器。通常，每个输出文件都用于为生成的对象构造容器。输入或输出文件的格式由通用对象部分的容器声明确定。

示例

要查看执行部分的示例，请参阅[规范文件的示例第 35 页码](#)。

属性（运行时）

此部分定义仅当执行节点时才可用的运行时属性集。

格式

其格式与元素定义主要部分中的 **Properties** 部分相似。有关详细信息，请参阅[第 58 页码Properties](#)。

当执行模型构建器节点或文档构建器节点时，将会创建一个**服务器临时文件**以存储模型输出或文档输出对象。服务器会访问此文件，并将其中的对象带到客户端以将其装入容器中。您需要在此指定此文件。

示例

本例说明如何指定服务器临时文件。

```
<Properties>
  <Property name="tempfile" valueType="string">
    <DefaultValue>
      <ServerTempFile basename="datatmp"/>
    </DefaultValue>
  </Property>
</Properties>
```

输入文件

本节定义要在执行节点前生成的输入文件集。此环境中的输入文件就是在服务器上执行节点时的输入文件。例如，执行节点时，模型填充器节点具有将传输到指定输入文件中的模型容器。

格式

```
<InputFiles>
  <ContainerFile id="identifier" path="path" container="容器">
    ...
</InputFiles>
```

在输入文件的 `ContainerFile` 元素中，其属性如下表所示。

表 4-4
容器文件属性 - 输入文件

特性	描述
id	容器文件的唯一标识符。
path	要在服务器中生成输入文件的位置（例如，服务器临时文件的位置 - 请参阅 属性（运行时）第 62 页码 ）。
container	容器（用于容纳要作为输入文件发送到服务器的对象）的标识符。

示例

```
<InputFiles>
  <ContainerFile id="pmml" path="{tempfile}.pmml" container="model"/>
</InputFiles>
```

输出文件

本节指定在服务器上执行节点时生成的输出文件。输出文件（例如，执行模型构建器节点或文档构建器节点的结果）将在执行后传输回客户端。

格式

```
<OutputFiles>
  <ContainerFile id="identifier" path="path" containerType="容器">
    ...
</OutputFiles>
```

在 `ContainerFile` 元素中，其属性如下表所示。

表 4-5
容器文件属性 - 输出文件

特性	描述
id	容器文件的唯一标识符。
path	要传输到客户端的对象在服务器上的位置（例如，服务器临时文件的位置 - 请参阅 属性（运行时）第 62 页码 ）。
containerType	对象的容器类型的标识符（例如，模型类型或文档类型的 ID），使对象以正确格式传输。 有关详细信息，请参阅第 43 页码容器类型 。

示例

```
<OutputFiles>
  <ContainerFile id="pmml" path="{tempfile}.pmml" containerType="mynode_model" />
  <ContainerFile id="htmloutput" path="{tempfile}.html" containerType="html_output" />
  <ContainerFile id="zipoutput" path="{tempfile}.zip" containerType="zip_outputType" />
</OutputFiles>
```

模块

本节指定执行节点时要用的服务器端共享库（例如，要加载到内存中的 DLL）。

格式

```
<Module libraryId="shared_library_identifier" name="node_name">
  <StatusCodes .../>
</Module>
```

其中：

`libraryId` 是在资源部分的 `Shared Library` 元素中声明的库的标识符。 [有关详细信息，请参阅第 39 页码共享库。](#)

如果库由多个节点共享，将使用 `name` 来标识正在执行的特定节点。如果只有一个节点在使用该库，名称字段可以为空。

示例

```
<Module libraryId="mynode1" name="mynode">
  <StatusCodes>
    <StatusCode code="0" status="error" message="An exception occurred" />
    <StatusCode code="1" status="error" message="Error reading input data" />
    ...
  </StatusCodes>
</Module>
```

状态代码

大多数程序都会执行某种错误检查操作并向用户显示任何必要的消息，通常会返回整数以表明成功完成状态或其他状态。服务器端 API 可以在执行包含节点的流后返回状态代码。 [有关详细信息，请参阅第 212 页码第 9 章中的状态详细信息文档。](#)

状态代码部分使您可以将一条消息与某个特定的状态代码关联，以便向用户显示。

格式

```
<StatusCodes>
  <StatusCode code="codenum" status="status" message="message_text"
    messageKey="message_key" />
  ...
</StatusCodes>
```

状态代码属性如下表所示。

表 4-6
状态代码属性

特性	描述
code	与消息关联的状态代码（一个整数值）。
status	状态分类： success - 执行节点成功 warning - 执行节点时发出警告 error - 执行节点不成功
message	当返回此状态代码时显示的消息。
messageKey	用于标识消息以便进行本地化。

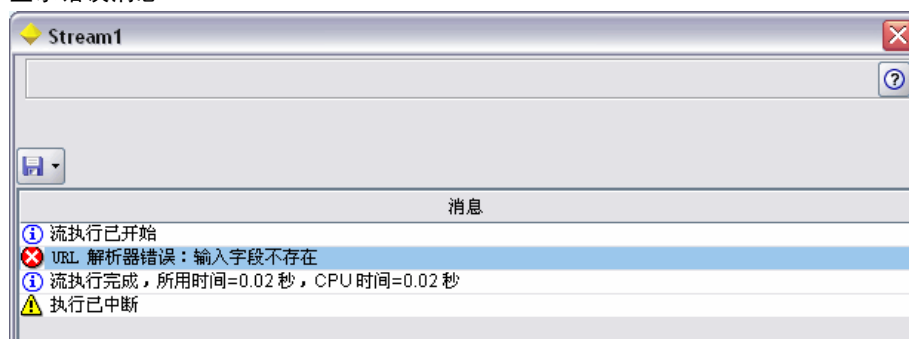
示例

在本例中，错误消息的文本包含在 `Statuscode` 元素中：

```
<StatusCodes>
  <StatusCode code="0" status="error" message="Cannot initialise a peer" />
  <StatusCode code="1" status="error" message="Error reading input data" />
  <StatusCode code="2" status="error" message="Internal Error" />
  <StatusCode code="3" status="error" message="Input Field Does Not Exist" />
</StatusCodes>
```

当执行时，如果服务器端 API 返回状态代码 3，将向用户显示以下内容：

图片 4-9
显示错误消息



在下一个示例中，错误消息的文本由 `messageKey` 属性引用：

```
<StatusCodes>
  <StatusCode code="0" status="error" messageKey="initErrMsg.LABEL" />
  <StatusCode code="1" status="error" messageKey="inputErrMsg.LABEL" />
  <StatusCode code="2" status="error" messageKey="internalErrMsg.LABEL" />
  <StatusCode code="3" status="error" messageKey="invalidMetadataErrMsg.LABEL" />
  ...
</StatusCodes>
```

属性文件（例如 `messages.properties`）位于规范文件所在的文件夹中，包含实际的消息文本以及其他显示文本：

```

...
initErrMsg.LABEL=Initialisation failed.
inputErrMsg.LABEL=Error when reading input data.
internalErrMsg.LABEL=Internal error.
invalidMetadataErrMsg.LABEL=Metadata (on input/output fields) not valid.
...

```

当需要对显示文本进行本地化以供海外市场使用时，此方法非常有用，因为所有要进行本地化的文本都可以在一个文件中找到。[有关详细信息，请参阅第 185 页码第 8 章中的本地化。](#)

输出数据模型

此元素仅在 Node 元素定义中使用。

输出数据模型部分指定某些特定属性如何影响数据模型。

输出数据模型可以采用以下三种方式之一确定：

- 使用规范文件中的字段集定义功能。[有关详细信息，请参阅第 77 页码字段集。](#)
- 使用实施可接收属性集和输入数据模型的数据模型提供程序界面、且返回一个数据模型实例的客户端 Java 类。
- 使用可接收属性集和输入数据模型、并返回一个元数据文档的服务器端共享库组件。

输出数据模型部分定义节点中的属性如何影响流过该节点的字段。输出数据模型可以：

- 保持输入数据模型不变
- 修改输入数据模型
- 使用另一个数据模型替换输入数据模型

例如，排序节点不会影响属性本身，但会对它们进行重新排序；导出节点可通过添加新字段修改数据模型；而汇总节点则完全替换数据模型。

在修改了输入数据模型的情况下，可以使用定义来添加新字段、或者修改或删除现有的字段。当替换数据模型时，只能添加新字段。规范文件支持这些基本操作（包括能够创建基于输入字段类型的新字段），以及循环访问输入字段集、关键字或表示输入字段集中的一组字段的列表属性。

格式

输出数据模型部分的通用格式如下所示，但有关这些情况的特定格式，请参阅[多字段选择器控件第 149 页码](#)和[单字段选择器控件第 157 页码](#)。

```

<OutputDataModel mode="mode" libraryId="container_name">
  -- data model operations --
</OutputDataModel>

```

输出数据模型属性如下表所示。

表 4-7
输出数据模型属性

特性	描述
mode	有关数据模型的效果： extend - 对现有模型添加一个新字段 fixed - 不更改 modify - 更改现有字段（例如删除或重命名） replace - 替换现有模型
libraryId	从中获取数据模型的服务器端容器的名称。

数据模型操作即添加新字段、修改或删除现有字段之类的操作。 [有关详细信息，请参阅第 72 页码数据模型操作。](#)

示例

`OutputDataModel` 元素包含在规范文件示例中。 [有关详细信息，请参阅第 35 页码规范文件的示例。](#)

Constructors

构造函数用于定义在流中执行节点时或生成返回流的对象时产生的对象。

有关如何对文件的这一部分进行编码的详细信息，请参阅从[使用构造函数第 109 页码](#)开始的章节。

通用特征

有些特征可以在规范文件的多个部分中使用，它们分别名为：

- 值类型
- 求值的字符串
- 操作
- 字段和字段元数据
- 字段集
- 角色
- 逻辑运算符
- 条件

值类型

值类型声明用于指定列、属性或属性类型规范可以采用的值类型。

字符串和加密字符串

`valueType="string"` 格式指定该值为文本字符串。如果用户输入的字段（例如密码字段）内容需要隐藏，`valueType="encryptedString"` 声明用于与这类字段相关的属性。

字段名

如果值以字段名的形式出现，请使用 `valueType="fieldName"` 格式。

数学、逻辑和日期表达式

如果值是一个数学（整数或双精度数）表达式、逻辑（true/false）表达式或日期表达式，请相应地将 `valueType` 设置为 `integer`、`double`、`boolean` 或 `date`。

枚举属性

枚举属性包含在 `valueType="enum"` 声明后紧跟的枚举部分中。 [有关详细信息，请参阅第 68 页码枚举属性。](#)

结构声明

`valueType="structure"` 声明用于指定包含其他已知属性的复合值。属性（Attribute）与属性（property）类似，但不能对属性（Attribute）进行构造化或键控。 [有关详细信息，请参阅第 69 页码结构化属性。](#)

- **键控指示器。** 指定属性是单个值还是散列表，散列表中的每个值都是指定类型的值。
- **值集合。** 指定如何确定可用值的集合。
- **键集合。** 对于键控属性，这用于指定如何确定可用键的集合。此信息也可用于向用户界面提供有关控制器最适用类型的提示。

数据库连接

用户可以使用这些连接字符串（例如 `user1@testdb`）登录数据库。但事先必须定义详细的数据库登录信息。 [有关详细信息，请参阅第 146 页码第 6 章中的数据库连接选择器控件。](#)

枚举属性

枚举属性就是可以从预定义值列表中获取值的属性。

格式

枚举属性的格式使用 `Enumeration` 部分，并在其中定义值列表，如下所示：

```
<PropertyTypes>
  <PropertyType id="identifier" valueType="enum">
    <Enumeration>
      <Enum value="value" label="display_label" labelKey="label_key"
        description="description" descriptionKey="description_key" />
    </Enumeration>
  </PropertyType>
</PropertyTypes>
```



```

    ...
  </Enumeration>
</PropertyType>
</PropertyTypes>

```

其中，PropertyType 属性包括：

- id 是属性类型的唯一标识符。
- valueType 指示属性的类型为枚举。

同时 Enum 属性包括：

- value（必需）是要在值列表中显示的属性值。
- label（必需）是当属性值显示在用户界面上时的显示名称。
- labelKey 用于标识标签以便进行本地化。
- description 是枚举值的说明。
- descriptionKey 用于标识说明以便进行本地化。

示例

```

<PropertyTypes>
  <PropertyType id="shared_enum1" valueType="enum">
    <Enumeration>
      <Enum value="value1" label="Value 5.1" labelKey="enum5.value1.LABEL" />
      <Enum value="value2" label="Value 5.2" labelKey="enum5.value2.LABEL" />
      <Enum value="value3" label="Value 5.3" labelKey="enum5.value3.LABEL" />
    </Enumeration>
  </PropertyType>
</PropertyTypes>

```

结构化属性

结构化属性就是在类似网格的结构（例如对话框中的表控件）中使用的属性。

格式

结构化属性的格式使用了一个定义结构的 Structure 部分，所以包含多个 Attribute 元素，如下所示：

```

<PropertyTypes>
  <PropertyType id="identifier" valueType="structure" isList="true_false">
    <Structure>
      <Attribute name="column_ID" valueType="value_type" isList="true_false" label="column_label"
        labelKey="label_key" defaultValue="value" description="description"
        descriptionKey="description_key" />
      ...
    </Structure>
  </PropertyType>
</PropertyTypes>

```

其中，PropertyType 元素属性包括：

- id 是属性类型的唯一标识符。
- valueType 指示属性为结构化类型。
- isList 指示该属性是特定值类型的一列值 (true) 还是单个值 (false)。

同时 Attribute 元素属性包含：

- name (必需) 是列的标识符。
- valueType 用于指定此列内容可以采用的值类型，包括：
 - string
 - encryptedString
 - integer
 - double
 - boolean
 - date
 - enum
- isList 指示该属性是特定值类型的一列值 (true) 还是单个值 (false)。这样，键控属性可以关联到固定的已知属性集 (例如，表示在特定字段上执行的不同汇总操作的 Boolean 属性) 或值列表 (例如，将字段名列表与某些其他字段名关联的值列表)。
- label (必需) 是列显示在用户界面上时的显示名称。
- labelKey 用于标识标签以便进行本地化。
- defaultValue 是当列显示时要出现在列中的值。
- description 是列的说明。
- descriptionKey 用于标识说明以便进行本地化。

示例 - 表控件

要查看如何在表控件中使用结构化属性的示例，请参阅[表控件第 160 页码](#)。

示例 - 键控属性类型

这些示例中的第一个示例说明了键控属性类型的使用，其中每个关联值为表示从固定操作集应用到某个字段的汇总操作的结构：

```
<PropertyType id="aggregateOps" isKeyed="true" valueType="structure">
  <Structure>
    <Attribute name="MIN" valueType="boolean" label="Min" />
    <Attribute name="MAX" valueType="boolean" label="Max" defaultValue="true"/>
    <Attribute name="SUM" valueType="boolean" label="Sum" defaultValue="false"/>
    <Attribute name="MEAN" valueType="boolean" label="Mean" defaultValue="false"/>
    <Attribute name="SDEV" valueType="boolean" label="SDev" defaultValue="false"/>
  </Structure>
</PropertyType>
```

因此，被声明使用 `aggregateOps` 属性类型的属性可以是：

```
<Property name="aggregationSettings" scriptName="aggregation_settings" type="aggregateOps"/>
```

这里属性包含多个值，每个值具有不同的关键字。例如，关键字 `name` 为字段名称（MIN、MAX 等）。

在接下来的键控属性类型示例中，每个关联值为包含单个属性的结构。在本例中，属性为表示应用到某个字段的乘数的双精度表达式列表：

```
<PropertyType id="multiplierOps" isKeyed="true" valueType="structure">
  <Structure>
    <Attribute name="multipliers" valueType="double" isList="true"/>
  </Structure>
</PropertyType>
```

被声明使用 `multiplierOps` 属性类型的属性可以是：

```
<Property name="multiplierSettings" scriptName="multiplier_settings" type="multiplierOps"/>
```

默认值

`DefaultValue` 元素用于指定服务器临时目录、文件或二者。它们的作用是存储模型输出对象或文档输出对象。

格式

```
<DefaultValue>
  <ServerTempDir basename="name"/>
  <ServerTempFile basename="name"/>
</DefaultValue>
```

其中，`basename`（必需）是临时目录或文件的名称。

示例

```
<DefaultValue>
  <ServerTempFile basename="datatmp"/>
</DefaultValue>
```

求值的字符串

在规范文件中声明的部分字符串可能会包含对属性名称的引用。这些字符串称为求值的字符串。

属性引用的语法为：

```
"${property_name}"
```

当访问求值的字符串时，所有属性引用都将替换为引用属性的值。如果属性不存在，将会发生错误。例如，当添加新字段时，节点定义中可能会有一个名为 `my_new_field` 的属性，而用户界面部分有一个控件，用户可以通过它来编辑这个属性的值。

示例

```
<AddField name="${my_new_field}" ... >
```

操作

规范文件的某些特定部分可支持多种不同的操作，例如添加字段、创建组件以及初始化属性。支持操作的部分包括：

- 输出数据模型（源节点和过程节点）
- 输入和输出数据模型（组件）
- 输出对象创建（模型构建器节点和文档构建器节点）
- 模型填充器创建（模型输出）

操作分为以下几种类型：

- 数据模型操作： `AddField`, `ChangeField`, `RemoveField`
- 迭代： `ForEach`

数据模型操作

您可以在数据模型中执行的操作包括：

- 在现有数据模型中添加新字段
- 修改数据模型中的现有字段
- 删除数据模型中的字段

添加字段

`AddField` 元素可用于在现有数据模型中添加新字段。

格式

```
<AddField prefix="prefix" name="name" direction="field_role" directionRef="field_role_ref"
  fieldRef="field_ref" group="group_id" label="label" missingValuesRef="mval_ref"
  storage="storage_type" storageRef="storage_ref" targetField="target_field" type="data_type"
  typeRef="type_ref" role="角色" tag="propensity_type" value="value" >
  <Range min="min_value" max="max_value" />
</AddField>
```

`AddField` 的属性如下所示。

表 4-8
AddField 属性

特性	描述
prefix	要对字段名称添加的前缀，例如用于表示模型输出字段。
name	(必需)要添加的字段的名称。这可以是硬编码字符串(例如 field8)或引用字段的求值字符串(例如 \${target})。有关详细信息，请参阅第 71 页码求值的字符串。
direction	字段的角色，例如字段是输入字段还是目标字段。in、out、both、partition 或 none 之一。有关详细信息，请参阅第 4 章中的设置字段角色中的 IBM SPSS Modeler 15 源、过程和输出节点。
direction-Ref	指定字段方向源自引用字段的求值字符串(例如 \${field1})所确定字段的方向。有关详细信息，请参阅第 71 页码求值的字符串。
fieldRef	指定所有引用值(directionRef、missingValuesRef、storageRef 和 typeRef)源自引用字段的求值字符串(例如 \${field1})所确定字段的相应值。有关详细信息，请参阅第 71 页码求值的字符串。
group	指定字段是字段组的成员。有关详细信息，请参阅第 93 页码第 5 章中的模型字段。
label	要添加的字段的标签。
missingValuesRef	指定缺失值处理方式源自引用字段的求值字符串(例如 \${field1})所确定字段的缺失值指定方式。有关详细信息，请参阅第 71 页码求值的字符串。
storage	字段值的数据存储类型 - integer、real、string、date、time、timestamp 或 unknown 之一。
storageRef	指定存储类型源自引用字段的求值字符串(例如 \${field1})所确定字段的存储类型。有关详细信息，请参阅第 71 页码求值的字符串。
targetField	对于模型输出字段，指定从中派生新字段数据的目标字段。这可以是硬编码字符串(例如 field8)或引用字段的求值字符串(例如 \${target})。有关详细信息，请参阅第 71 页码求值的字符串。
type	字段的数据类型 - auto、range、discrete、set、orderedSet、flag 或 typeless 之一。有关详细信息，请参阅第 4 章中的测量级别中的 IBM SPSS Modeler 15 源、过程和输出节点。
typeRef	指定数据类型源自引用字段的求值字符串(例如 \${field1})所确定字段的数据类型。有关详细信息，请参阅第 71 页码求值的字符串。
role	模型输出字段中所存储的数据类型 - unknown、predictedValue、predictedDisplayValue、probability、residual、standardError、entityId、entityAffinity、upperConfidenceLimit、lowerConfidenceLimit、propensity、value 或 supplementary 之一。有关详细信息，请参阅第 79 页码角色。
tag	只在 role 有值 propensity 的情况下使用；指明倾向类型，可以是 RAW，也可以是 ADJUSTED。
value	指定新字段要包含的值源自引用字段的求值字符串(例如 \${field1})所确定的字段。有关详细信息，请参阅第 71 页码求值的字符串。

Range 属性如下所示。

表 4-9
范围属性

属性	说明
最小值	字段可以接受的最小值。
最大值	字段可以接受的最大值。

示例

以下示例添加一个名为 `field8` 的字符串字段：

```
<AddField name="field8" storage="string" />
```

下一个示例将演示当添加字段时如何使用对属性名称的引用。此处为字段添加的名称与先前定义的属性 `prop1` 的值相匹配：

```
<AddField name="{prop1}" ... />
```

在下面的示例中，如果目标字段名为 `field1`，模型将创建一个名为 `$$-field1` 的输出字段来存储 `field1` 的预测值：

```
<AddField prefix="$$" name="{target}" role="predictedValue" targetField="{target}"/>
```

下一个示例添加的模型输出字段用于存储介于 0.0 和 1.0 之间的概率得分：

```
<AddField prefix="$$C" name="{target}" storage="real" role="probability" targetField="{target}">
  <Range min="0.0" max="1.0"/>
</AddField>
```

在最后的示例中，为每个模型输出字段添加一个输出字段，用于存储介于 0.0 和 1.0 之间的概率得分，其值源自变量 `fieldValue` 的值：

```
<ForEach var="fieldValue" inFieldValues="{field}">
  <AddField prefix="$$SP" name="{fieldValue}" storage="real" role="probability" targetField="{field}"
    value="{fieldValue}">
    <Range min="0.0" max="1.0"/>
  </AddField>
</ForEach>
```

有关详细信息，请参阅第 71 页码求值的字符串。

更改字段

`ChangeField` 元素可用于修改数据模型中的现有字段。

格式

```
<ChangeField
name="name" fieldRef="field_reference" direction="field_role" storage="storage_type" type="data_type" >
  <Range min="min_value" max="max_value" />
</ChangeField>
```

`ChangeField` 的属性如下所示。

表 4-10
ChangeField 属性

特性	描述
name	(必需) 要更改的字段名称。
fieldRef	字段的参考值。

特性	描述
direction	字段的角色，例如字段是输入字段还是目标字段。in、out、both、partition 或 none 之一。有关详细信息，请参阅第 4 章中的设置字段角色中的 IBM SPSS Modeler 15 源、过程和输出节点。
storage	字段值的数据存储类型 - integer、real、string、date、time、timestamp 或 unknown 之一。
type	字段的数据类型 - auto、range、discrete、set、orderedSet、flag 或 typeless 之一。有关详细信息，请参阅第 4 章中的测量级别中的 IBM SPSS Modeler 15 源、过程和输出节点。

Range 属性如下所示。

表 4-11
范围属性

属性	说明
最小值	字段可以接受的最小值。
最大值	字段可以接受的最大值。

删除字段

RemoveField 元素可用于删除数据模型中的字段。

格式

```
<RemoveField fieldRef="field_reference" />
```

其中，fieldRef 是字段的参考值。

使用 ForEach 元素进行迭代

在某些位置，采用重复执行相同操作的方式处理一组值中的每个值非常有用。规范文件支持一种简单的 ForEach 迭代器，它会按顺序将一个临时属性与所提供数据组中的每个值绑定。可以将 ForEach 循环设置为以下几种迭代方式之一：

- 在两个具有可选步长的整数值之间
- 在列表属性的值之间
- 在键控属性的键之间
- 在字段组的字段之间

格式

```
<ForEach var="field_name" from="integer_exp" to="integer_exp" step="integer_exp"
inFields="字段" inFieldValues="field_name" inProperty="property_name" >
  -- data model operation --
</ForEach>
```

其中：

var（必需）指定的字段包含将对其应用迭代的值。

`from` 和 `to` 指定的整数（或者求值结果为整数的表达式）表示迭代的下限和上限，同时使用可选属性 `step` 表示整数步长。

`inFields`、`inFieldValues` 和 `inProperty` 是 `from/to/step` 格式的替代选项：

- `inFields` 指定用于执行迭代的字段集，包括：
 - `inputs` - 节点的输入字段
 - `outputs` - 来自节点的输出字段
 - `modelInput` - 在模型签名中指定的输入字段
 - `modelOutput` - 在模型签名中指定的输出字段
- `inFieldValues` 指定字段名（或者表示字段名的属性）并迭代该字段的元数据中的所有值
- `inProperty` 指定对其执行迭代的属性的名称

可以在 `ForEach` 元素中指定的数据模型操作是 `AddField`、`ChangeField` 或 `RemoveField` 元素之一。[有关详细信息，请参阅第 72 页码数据模型操作。](#) `ForEach` 元素也可以嵌套。

示例

以下示例将执行一个操作十次：

```
<ForEach var="val" from="1" to="10">
  ...
</ForEach>
```

以下示例将按整数属性指定的次数执行操作：

```
<ForEach var="val" from="1" to="{history_count}">
  ...
</ForEach>
```

在下一个示例中，将对节点的输出字段中的所有值进行迭代：

```
<ForEach var="field" inFields="outputs">
  ...
</ForEach>
```

下面的示例指定字段为由 `{field}` 标识的字段，并对该字段元数据中的所有值进行迭代：

```
<ForEach var="fieldValue" inFieldValues="{field}">
  ...
</ForEach>
```

下一个示例将对列表属性中的所有值进行迭代：

```
<ForEach var="val" inProperty="my_list_property">
  ...
</ForEach>
```

下面的示例将对键控属性中的所有键值进行迭代：


```
<ForEach var="key" inProperty="my_keyed_property">
  ...
</ForEach>
```

字段和字段元数据

节点、模型和数据源作为**数据模型提供程序**，它们可以定义可从其他对象访问的字段元数据。

数据模型提供程序具有一个输入数据模型和一个输出数据模型。输出数据模型可以按照输入数据模型的方式定义，例如当通过添加字段扩展输入模型时，或者当更改现有模型时。

这些对象各自的要求稍有不同。

节点。输入数据模型可以被引用，但不能对其进行修改。输出数据模型可以基于输入数据模型，或者也可以替换它。只要节点属性或输入数据模型发生更改，都将重新计算输出数据模型。模型填充器节点的输出数据模型也可以引用模型组件的输出数据模型。

模型。默认情况下，输入和输出数据模型（模型签名）均基于创建模型时使用的输入和输出字段设置。理想状态下，模型构建过程将返回一个用于定义所需输入字段和已生成输出字段的元数据文件。定义后，模型签名将无法更改。但是，使用模型填充器节点中的属性可以修改来自填充器节点的数据模型输出。例如，这些属性可以定义某个聚类 ID 是作为字符串还是作为整数返回，或者要生成多少顺序 ID。此外，模型签名通常将输出的字段角色（`direction`）指定为"out"，而节点却似乎以字段角色 "in" 生成它们。

数据源。在数据读取器节点中使用的数据源可以指定一个输出数据模型。输入数据模型始终为空。

字段集

可以在多个位置使用字段集以从数据模型提供程序中选择字段子集。数据模型提供程序既可以是宿主类对象，也可以是宿主类对象的容器。字段过滤器的初始状态可以先包含所有可用字段，然后排除特定类型的字段；也可以先为空的字段集，然后再包含所需字段或添加新字段。

下面的示例将演示扩展节点如何指定输出数据模型。关键字段由一个名为 `keys` 的列表属性指定，该列表属性后跟随一个可选的可生成记录计数字段，该字段的名称也是由属性指定。

```
<OutputDataModel mode="replace">
  <ForEach var="field" inProperty="keys">
    <AddField name="{field}" fieldRef="{field}"/>
  </ForEach>
  <AddField name="{record_count_name}" storage="integer">
    <Condition property="include_record_count" op="equals" value="true"/>
  </AddField>
</OutputDataModel>
```

字段集和模型构建

下面的示例将演示模型填充器如何使用先前创建的模型组件中的信息生成自己的输出字段：

```
<OutputDataModel mode="modify">
  <AddField provider="model" dataModel="output">
</OutputDataModel>
```

AddField 和 **ForEach** 均用于指定数据模型提供程序，同时还可以指定应该使用哪些输入或输出数据模型。它们提供了一种用于指定来自数据模型提供程序的字段集（或子集）的机制。默认的提供程序为表示宿主类元素（即不是所创建的对象）的 **this**，且具有默认情况下使用的输入字段集。如果没有指定字段集，将使用所有可用字段。

字段集可以基于存储、类型、字段角色或名称。如果字段集基于名称，则需要对列表属性的引用。字段集可以为满（默认），也可以为空；前者允许将某些字段排除在外，而后者则允许将某些字段包含在内。可以为每个单独的过滤器指定多个值，这些值将作为“intersection”或“and”运算符使用，例如：

```
<FieldSet include="none">
  <Include direction="in" storage="string"/>
</FieldSet>
```

此示例由空字段集开始（由 **include="none"** 指定），然后将包含字段角色（**direction**）为 **"in"** 以及存储类型为字符串的字段。

又例如：

```
<FieldSet include="all">
  <Exclude type="typeless"/>
</FieldSet>
```

此示例包含所有可用字段（由 **include="all"** 属性指定，这是默认的行为），然后排除任何具有 **typeless** 类型的字段。这将包含将 **direction** 设置为 **"in"** 或 **"both"** 的字段。

也可指定多个过滤器，它们将作为“union”或“or”运算符使用，例如：

```
<FieldSet include="all">
  <Exclude type="discrete" storage="real"/>
  <Exclude type="discrete" storage="integer"/>
</FieldSet>
```

此示例排除了那些具有离散实数存储或离散整数存储的字段。

请注意，当在初始为空的字段集中放入字段时，**include** 语句的顺序通常不会影响字段的放入顺序。也就是说，对于来自字段集提供程序的字段，将以它们的自然顺序针对每个条件进行计算，以确定是否应将它们放入字段集中。

角色

角色用于描述数据模型输出字段中所存储的数据的类型。角色可以由 `AddField` 元素指定，并由 `Condition` 元素进行测试。

可能的角色如下所示。

表 4-12
模型输出角色

角色	含义
unknown	未指定角色（不应发生）。
predictedValue	此字段包含目标字段的预测值。
probability	预测的概率或置信度。
residual	残差值。
standardError	预测的标准误。
entityId	实体 ID，通常表示聚类模型中的聚类 ID。
entityAffinity	实体仿射性，通常表示与模型聚类中心的距离。
upperConfidenceLimit	预测的置信上限。
lowerConfidenceLimit	预测的置信下限。
propensity	倾向得分。另一个 <code>tag</code> 属性指定这是指原始倾向还是指调整倾向。
value	用于表示与另一个输出相关联的模型的值（请参阅下文）。
supplementary	由其他角色不涉及的模型生成的信息。

作为 `value` 的示例，异常检测模型生成了多组字段，其中每组包含两个字段，一个表示字段名称，另一个用于指定衡量该字段异常程度的尺度。在本例中，`value` 将为字段名。

逻辑运算符

许多元素都可以使用逻辑运算符 `And`、`Or` 和 `Not` 来指定各种不同的处理操作，例如当设置复合条件时（请参阅[复合条件第 83 页码](#)）。

格式

`And` 元素的格式如下所示。`Or` 和 `Not` 元素的格式几乎相同，唯一区别是它们的封入标记分别为 `<Or>...</Or>` 和 `<Not>...</Not>`。

```
<And>
  <Condition .../>
  <And .../>
    <Or .../>
    <Not .../>
</And>
```

`Condition` 元素用于指定要进行测试的条件。[有关详细信息，请参阅第 80 页码条件。](#)

请注意，**And**、**Or** 和 **Not** 子元素可以是嵌套元素。

条件

某些对象的行为可以通过使用由 **Condition** 元素（相当于 IF 语句）指定的条件进行修改。例如，由命令执行的节点可以在执行信息中添加条件，例如如果属性具有某个特定值，则仅包含一个特定选项。类似地，用户界面中的某个属性控件可能仅在另一个控件具有特定值时才会启用或可视。

条件既可以是简单条件，也可以是复合条件。**简单条件**由下列部分组成：

- 值源（可以是属性或控件）
- 测试
- 可选的测试值

复合条件允许其他条件进行组合，从而形成复杂的逻辑条件。复合条件可以使用：

- **And**
- **Or**
- **Not**

格式

```
<Condition container="container_name" control="prop_name" property="name" op="operator" value="value" />
```

其中：

container 用于指定该条件将对其值进行测试的特定容器的名称。

control 用于指定该条件将对其值进行测试的属性控件。**prop_name** 就是在其中定义控件的元素（例如，在某个对话框选项卡的属性面板中）的 **property** 属性值。

property 用于指定该条件将对它的值进行测试的属性。**name** 是要在其中定义属性的 **Property** 元素的 **name** 属性值。

op 是条件运算符。 [有关详细信息，请参阅第 80 页码条件运算符。](#)

value 将由条件进行测试的特定值。

示例

有关条件设置的示例，请参阅[简单条件第 83 页码](#)和[复合条件第 83 页码](#)。

条件运算符

一组可用于满足大多数条件的运算符。

表 4-13
任意值支持的测试

运算符	值	描述
equals	value	如果属性等于所提供的值，则为真（字符串值需要区分大小写）。
notEquals	value	如果属性不等于所提供的值，则为真（字符串值需要区分大小写）。
in	值列表	如果属性在所提供的值列表中，则为真。

表 4-14
数字值支持的测试

运算符	值	描述
lessThan	number	如果属性小于所提供的数字，则为真。
lessOrEquals	number	如果属性小于或等于所提供的数字，则为真。
greaterThan	number	如果属性大于所提供的数字，则为真。
greaterOrEquals	number	如果属性大于或等于所提供的数字，则为真。

表 4-15
字符串值支持的测试

运算符	值	描述
isEmpty	-	如果属性具有长度为零的字符串，则为真。
isNotEmpty	-	如果属性具有长度非零的字符串，则为真。
startsWith	string	如果属性由所提供的字符串开头，则为真（区分大小写）。
startsWithIgnoreCase	string	如果属性由所提供的字符串开头，则为真，不区分大小写。
endsWith	string	如果属性由所提供的字符串结尾，则为真（区分大小写）。
endsWithIgnoreCase	string	如果属性由所提供的字符串结尾，则为真，不区分大小写。
equalsIgnoreCase	string	如果属性等于所提供的字符串，则为真，不区分大小写。
hasSubstring	string	如果属性包含所提供的字符串，则为真（区分大小写）。
hasSubstringIgnoreCase	string	如果属性包含所提供的字符串，则为真，不区分大小写。
isSubstring	string	如果属性是所提供字符串的子字符串，则为真（区分大小写）。
isSubstringIgnoreCase	string	如果属性是所提供字符串的子字符串，则为真，不区分大小写。

表 4-16
列表属性支持的测试

运算符	值	描述
isEmpty	-	如果列表中的项目数为 0，则为真。
isNotEmpty	-	如果列表中的项目数不为 0，则为真。

运算符	值	描述
countEquals	number	如果列表中的项目数等于所提供的值，则为真。
countLessThan	number	如果列表中的项目数小于所提供的值，则为真。
countLessOrEquals	number	如果列表中的项目数小于或等于所提供的数字，则为真。
countGreaterThan	number	如果列表中的项目数大于所提供的数字，则为真。
countGreaterOrEquals	number	如果列表中的项目数大于或等于所提供的数字，则为真。
contains	value	如果列表中含有所提供的项，则为真。

表 4-17
字段属性支持的测试

运算符	描述
storageEquals	如果存储等于所提供的值，则为真。
typeEquals	如果类型等于所提供的值，则为真。
directionEquals	如果字段角色（direction）等于所提供的值，则为真。
isMeasureDiscrete	如果字段数据类型为 discrete（即只能是 set、flag 或 orderedSet 之一），则为真。
isMeasureContinuous	如果字段数据类型为 range，则为真。
isMeasureTypeless	如果字段数据类型为 typeless，则为真。
isMeasureUnknown	如果字段数据类型为 unknown，则为真。
isStorageString	如果字段存储类型为 string，则为真。
isStorageNumeric	如果字段存储类型为 numeric，则为真。
isStorageDatetime	如果字段存储类型为 datetime，则为真。
isStorageUnknown	如果字段存储类型为 unknown，则为真。
isModelOutput	如果字段为模型输出字段，则为真。
modelOutputRoleEquals	如果字段的角色为下表中显示的有效角色之一，则为真。
modelOutputTargetFieldEquals	如果目标字段等于指定值（字符串），则为真。
modelOutputHasValue	如果字段为模型输出字段并且具有相关值，则为真。
modelOutputTagEquals	如果标记等于指定值（字符串），则为真。

支持键控属性的条件运算符包括：

- isEmpty
- isEmpty
- countEquals
- countLessThan
- countLessOrEquals
- countGreaterThan
- countGreaterOrEquals

■ contains

简单条件

简单条件包含要测试的初始值源（可以是属性或控制器名称，或者是求值表达式）、要执行的测试以及执行测试时要对比的值（可选）。

示例

如果名为 `values_grouped` 的布尔值属性为真，则下列示例的值也将为真：

```
<Condition property="values_grouped" op="equals" value="true"/>
```

在下面的示例中，如果已选中显示布尔值的 `values_grouped` 控件，则值为真：

```
<Condition control="values_grouped" op="equals" value="true"/>
```

在下面的示例中，如果名为 `plot_fields` 的列表属性包含至少一个值，则值为真：

```
<Condition property="plot_fields" op="countGreaterThan" value="0"/>
```

在下面的示例中，如果名为 `input_fields` 的列表属性包含的值均已实例化，则值为真：

```
<Condition property="input_fields" op="instantiated" listMode="all"/>
```

在最后这个示例中，如果名为 `input_fields` 的列表属性至少具有一个表示非实例化字段的值，则值为真：

```
<Condition property="input_fields" op="uninstantiated" listMode="any" />
```

复合条件

您可以使用逻辑运算符将多组简单条件组合在一起。

示例

在下面的示例中，如果布尔值的 `values_grouped` 属性为真并且 `group_fields` 至少包含一个值，则值为真：

```
<And>  
  <Condition property="values_grouped" op="equals" value="true"/>  
  <Condition property="group_fields" op="countGreaterThan" value="0"/>  
</And>
```

在下面的示例中，如果布尔值的 `values_grouped` 属性为真或者如果 `group_fields` 至少包含一个值，则值为真：

```
<Or>  
  <Condition property="values_grouped" op="equals" value="true"/>  
  <Condition property="group_fields" op="countGreaterThan" value="0"/>  
</Or>
```

在下面的示例中，如果 `group_fields` 至少包含一个值，则值为真：

```
<Not>
  <Condition property="group_fields" op="equals" value="0"/>
</Not>
```

对复合条件进行嵌套可以提供任意条件组合。

在脚本中使用 CLEF 节点

可以通过 `Node` 元素的 `scriptName` 属性，在脚本中引用 CLEF 节点。同样，通过 `Property` 元素的 `scriptName` 属性，可以在脚本中引用节点的属性。

在这两种情况下，`scriptName` 属性是可选的，不过我们建议使用此属性，以避免扩展或属性间发生名称冲突。

如果在节点定义中略去脚本名称，则脚本可以通过以扩展名称为前缀的 `id` 属性的值来引用节点。例如，假定有一个名为 `myext` 的扩展，它使用 ID `import` 定义数据读取器节点，脚本就可以使用 `myextimport` 引用此节点。

如果在属性定义中略去脚本名称，则脚本可以通过其 `name` 属性的值来引用属性。

有关更多信息，请参阅《IBM® SPSS® Modeler脚本编写与自动化指南》。

示例 - 编辑和执行节点

以下示例将演示如何使用脚本来自动完成编辑和执行 [数据读取器节点 \(Apache Log Reader\)](#) 第 30 页码 中所示例数据读取器节点的任务。

在 Apache Log Reader 节点的规范文件中，节点规范的开头为：

```
<Node
id="apache_logreader" type="dataReader" palette="import" labelKey="apacheLogReader.LABEL">
  <Properties>
  <Property name="log_filename" valueType="string" labelKey="logfileName.LABEL" />
  </Properties>
```

在脚本中，可以像下面这样引用节点和属性：

```
createapache_logreader
set :apache_logreader.log_filename='installation_directory\Demos\combined_log_format.txt'
create tablenode at 200 100
connect :apache_logreader to :tablenode
execute :tablenode
```

其中，`installation_directory` 是 IBM® SPSS® Modeler 的安装目录。

运行此脚本：

- 创建数据读取器节点
- 指定 `combined_log_format.txt` 作为要读取的 Apache 日志文件

- 创建表节点
- 将数据读取器节点连接到表节点
- 执行表节点

示例 - 键控属性

键控属性支持标准脚本编写语法。例如，[结构化属性第 69 页码](#) 中的第一个键控属性类型示例所示的结构在脚本中可以定义如下：

```
set :mynode.aggregation_settings.Age = {true true false false false}
```

单个属性可以修改如下：

```
set :mynode.aggregation_settings.Age.MIN = true
```

保持向后兼容性

当对现有扩展计划更新时，请注意保持与先前发布版本的扩展兼容。有些更改将不会产生不利的影响，有些更改涉及极大的风险，还有一些更改则会破坏兼容性，应避免此类更改。

无风险的更改

下列更改将不会影响向后兼容性：

- 添加新 `Node`、`ModelOutput`、`DocumentOutput` 或 `InteractiveModelBuilder` 元素
- 对这些元素添加新的 `Property` 定义及其相关的新控件
- 对这些元素添加新容器*
- 对现有枚举属性添加新值

*请注意，任何使用这些新容器的代码都应允许以下事实，即对于使用早期版本的扩展创建的对象，这些容器将为空。

包含极大风险的更改

对现有声明的更改包含破坏兼容性的巨大风险。这些更改应进行仔细测试后再发布。

需要避免的更改

下面这些更改是已知的会破坏兼容性的更改，应该避免：

- 更改 `ExtensionDetail` 元素中的 `id` 或 `providerTag` 属性的值。
- 更改 `Node`、`ModelOutput`、`DocumentOutput` 或 `InteractiveModelBuilder` 元素中的 `id` 值
- 从扩展删除一个 `Node`、`ModelOutput`、`DocumentOutput` 或 `InteractiveModelBuilder` 元素
- 更改 `Property` 或 `PropertyType` 元素中的 `valueType` 属性的值

构建模型和文档

模型和文档构建简介

标准的 IBM® SPSS® Modeler 模块包括允许用户生成（或“构建”）多种模型和图形的节点。通过 CLEF，您可以定义其他节点以构建未作为标准提供的其他模型和文档（图形和报告）。

定义模型构建器或文档构建器节点时，还需要定义执行这些节点时所生成的对象。您可以通过名为“构造函数”的项来执行此操作。

以下几节将详细介绍此过程。

模型

模型是一组规则、一个公式或一个方程，可用于根据输入字段集预测结果。预测结果的能力是预测性分析的核心目标。在 IBM® SPSS® Modeler 中，您可以通过以下方式实现此目标：

- 根据现有数据生成模型
- 将生成的模型应用到数据中以执行预测

生成模型的过程也称为“构建”模型，在 SPSS Modeler 中，您可以通过建模节点来实现此过程。在 CLEF 中，建模节点被称为**模型构建器节点**，该名称是从用于定义这些节点的 XML 语句的语法中派生而来的。[有关详细信息，请参阅第 11 页码第 2 章中的模型构建器节点。](#)

将模型应用到数据的过程称为“评分数据”。这样您便可以使用从构建的模型中获得的信息来为新记录做出预测。在 SPSS Modeler 中，可以通过将生成的模型图标添加到流工作区中实现此操作。图标采用金色块状形式，因此在 SPSS Modeler 中所生成的模型称为“模型块”。在 CLEF 中，管理器窗格“模型”选项卡上的模型块称为**模型输出对象**，当将它添加到工作区中时，称为**模型填充器节点**。[有关详细信息，请参阅第 13 页码第 2 章中的模型填充器节点。](#)

文档

在有些情况下，您可能希望生成对象而不是模型，例如图形或报告输出。在 IBM® SPSS® Modeler 中，我们将这些对象称为**文档**，它们可通过**文档构建器节点**生成。[有关详细信息，请参阅第 12 页码第 2 章中的文档构建器节点。](#)

Constructors

构造函数用于定义在流中执行节点时或生成返回流的对象时产生的对象。

可以为以下任意内容定义构造函数：

- 模型构建器节点

- 文档构建器节点
- 模型填充器节点
- 模型输出对象

对于**模型构建器**或**文档构建器**节点，构造函数使这些节点能够定义在执行节点时生成输出对象的方式。输出对象定义可以包含多个属性和组成部分，而构造函数部分定义如何初始化或通过执行生成的对象创建这些属性和组成部分。

对于**模型填充器**节点，构造函数定义节点可以生成回流或“模型”选项卡的对象的类型。

对于为**模型输出对象**定义的构造函数，它们可以：

- 指定在将模型输出对象拖放到流工作区时创建的模型填充器节点
- 通过用于创建模型输出对象的设置生成模型构建器节点

有关详细信息，请参阅第 109 页码使用构造函数。

构建模型

指定可以从中生成模型的节点（即，模型构建器节点）时，需要定义节点与 IBM® SPSS® Modeler 的模型构建器组件通信的方式，这就是实际创建模型的过程。您可以在规范文件的 **Node** 元素的定义中执行此操作。

除非另行指定，否则只要最终用户单击“模型构建器节点”对话框的**执行按钮**便开始构建模型。但是，也可以定义**交互模型**，通过它最终用户可以在单击执行之后在实际构建模型之前精练或修改数据值。此外，构建交互模型还需要包含可以定义交互的特定元素。有关详细信息，请参阅第 96 页码构建交互模型。

定义模型构建器节点时，**Node** 元素必须包含：

- **type="modelBuilder"** 属性
- **ModelBuilder** 子元素
- 包含 **CreateModelOutput** 元素的 **Constructors** 子元素（请参阅使用构造函数第 109 页码）

有关 **Node** 元素规范的格式信息，请参阅节点第 54 页码。

注意：在后续章节的元素定义（通常由标题**格式**标识）部分，除非指示为“（必需）”，否则元素属性和子元素均为可选项。要了解完整的元素语法信息，请参阅附录 A、CLEF XML 架构第 224 页码。

如果要构建模型，扩展还需要 **ModelOutput** 元素来描述生成的模型（请参阅模型输出第 95 页码）。**ModelOutput** 元素需要包含具有 **CreateModelApplier** 定义的 **Constructors** 子元素。有关详细信息，请参阅第 112 页码创建模型填充器。

模型构建器

ModelBuilder 元素定义模型构建器节点的行为。可通过元素属性和一个或多个子元素来实现此操作。

格式

```
<ModelBuilder
allowNoInputs="true_false" allowNoOutputs="true_false" nullifyBlanks="true_false"
  miningFunctions="[function1 function2 ...]" >
  <Algorithm .../>
  <ModelingFields .../>
  <ModelGeneration .../>
  <ModelFields .../>
  <AutoModeling .../>
</ModelBuilder>
```

其中：

- `allowNoInputs` 和 `allowNoOutputs` 必须分别显式使用，才能构建没有输入字段或输出字段的模型。
- `nullifyBlanks`，如果设置为 `false`，则禁用在传递到 IBM® SPSS® Modeler 的模型构建器组件的数据中用空值（由 `$null$` 表示）替换空白值的功能。默认情况下将使用空值替换空白，但您可能希望禁用此功能，例如，如果您的算法需要区别对待空白和空值。
- `miningFunctions`（必需）标识数据挖掘功能或模型执行功能。

表 5-1
数据挖掘功能

函数	描述
分类	预测具有已知目标值的记录的未知目标属性的离散（即， <code>set</code> 、 <code>flag</code> 或 <code>orderedSet</code> 数据类型）值。
<code>approximation</code>	预测具有已知目标值的记录的未知目标属性的连续（即， <code>range</code> 数据类型）值。
聚类	标识相似记录的组并相应地进行标记。
<code>association</code>	标识数据中相关的事件或属性。
<code>sequence</code>	在按时间建立结构的数据中搜索顺序模式。
<code>reduction</code>	降低数据的复杂性，例如，通过汇总原始数据字段内容的派生字段。
<code>conceptExtraction</code>	在文本挖掘中使用。
<code>categorize</code>	在文本挖掘中使用。
<code>timeSeries</code>	根据过去数据中的模式预测未来值。
<code>anomalyDetection</code>	根据与聚类组正常值的偏差搜索异常值。
<code>attributeImportance</code>	标识对目标属性影响最大的属性。
<code>supervisedMultiTarget</code>	为多个概率中的某个概率估计一个（ <code>yes</code> 或 <code>no</code> ）结果的似然。

如果模型执行多个功能，则在方括号中用空格分隔功能名称，如下例所示：

```
<ModelBuilder miningFunctions="[classification approximation]">
...
</ModelBuilder>
```

子元素

`ModelBuilder` 元素的子元素如下：

表 5-2
模型构建器声明的子元素

子元素	描述	请参阅...
算法	(必需) 指定用于生成模型的算法。	算法第 89 页码
ModelingFields	指定随后在“用户界面”部分用于定义模型的输入和输出字段控件的位置的标识符。控件本身在 ModelingFields 的 InputFields 和 OutputFields 子元素中定义。	建模字段第 89 页码
ModelGeneration	指定随后在“用户界面”部分用于为生成的模型定义模型名称控件的位置的标识符。	模型生成第 93 页码
ModelFields	指定使用此模型评分数据的输入和输出字段组。	模型字段第 93 页码
AutoModeling	允许此模型被整体建模节点使用, 例如自动分类器、自动聚类或自动数值。	自动建模第 101 页码

算法

Algorithm 元素可定义用于生成模型的算法的详细信息。

```
<Algorithm value="model_output_id" label="display_label" labelKey="label_key"/>
```

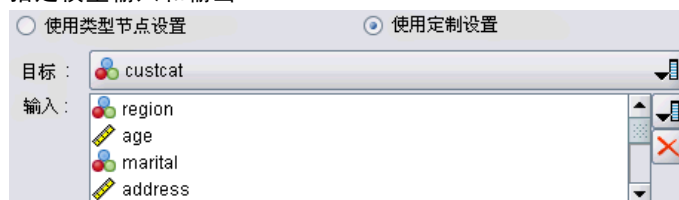
其中:

- value (必需) 是算法的内部名称。规范文件中的很多其他地方均可引用此名称。
有关详细信息, 请参阅第 94 页码模型构建器示例。
- label (必需) 是算法的描述。
- labelKey 标识标签以便进行本地化。

建模字段

采用“类型”节点是指定模型输入和输出字段的标准方式。用户可根据需要将字段角色设置为 in 或 out。(可选) 对于模型构建器节点, 可以允许用户选择覆盖上游“类型”节点中的设置和使用自定义设置, 例如:

图片 5-1
指定模型输入和输出



通过 ModelingFields 元素可以实现此操作。此元素指定随后在模型构建器节点声明的用户界面部分用于定义模型的输入和输出字段的控件的位置的标识符。控件本身通过 InputFields 和 OutputFields 子元素定义。

格式

```
<ModelingFields controlId="control_identifier" ignoreBOTH="true_false" >
  <InputFields .../>
  <OutputFields .../>
</ModelingFields>
```

其中：

- **controlId**（必需）是随后在模型构建器节点声明的“用户界面”部分的 **SystemControls** 元素中使用的标识符。这样便可以标识节点对话框的哪个选项卡将包含模型的输入和输出字段控件。
- **ignoreBOTH**，如果设置为 **true**（默认），则指定模型忽略字段角色设置为 **both** 的字段。

InputFields 和 **OutputFields** 元素将在 [输入字段第 90 页码](#) 及其后的章节中介绍。

示例

本示例阐释了模型构建器节点对话框的“字段”选项卡中的一组建模字段控件的用法。首先，为该控件组指定标识符：

```
<ModelBuilder miningFunctions="[classification]">
...
  <ModelingFields controlId="modelingFields">
    <InputFields property="inputs" onlyNumeric="true" multiple="true" label="Inputs"
      labelKey="inputFields.LABEL"/>
    <OutputFields property="target" multiple="false" types="[set flag]" label="Target"
      labelKey="targetField.LABEL"/>
  </ModelingFields>
...
</ModelBuilder>
```

modelingFields 标识符随后将在节点对话框的“用户界面”部分被引用，在定义“字段”选项卡的位置处：

```
<UserInterface ...>
  <Tabs defaultTab="1">
    <Tab label="Fields" labelKey="Fields.LABEL" helpLink="modeling_fieldstab.htm">
      <PropertiesPanel>
        <SystemControls controlId="modelingFields">
          </SystemControls>
      </PropertiesPanel>
    </Tab>
  </Tabs>
...
</UserInterface>
```

输入字段

InputFields 元素定义用户可以为模型选择一个或多个输入字段（即预测变量）的一组字段。

此设置组成了可在此节点中看到的所有字段。如果字段从此节点上游进一步过滤，那么只有通过过滤器的字段才是可见的。该列表可以进一步对之进行限制，其方式是指定仅可选择特定存储和数据类型的字段。

```
<InputFields storage="storage_types" onlyNumeric="true_false" onlySymbolic="true_false"
  onlyDatetime="true_false" types="data_types" onlyRanges="true_false" onlyDiscrete="true_false"
  property="property_name" multiple="true_false" label="标签" labelKey="label_key"/>
```

通过指定两个属性（其中一个属性必须来自于下面的列表），可以将要使用的字段列表限制为输入字段：

- **storage** 是列表属性，该属性指定在列表中所允许的字段的存储类型，例如，**storage="[integer real]"** 表示只列出上述存储类型的字段。有关可能存储类型的设置的信息，请参阅数据和存储类型第 202 页码下的表格。
- **onlyNumeric** 如果设置为 **true**，则指定只列出存储类型为数字的字段。
- **onlySymbolic** 如果设置为 **true**，则指定只列出存储类型为符号（即字符串）的字段。
- **onlyDatetime** 如果设置为 **true**，则指定只列出存储类型为日期和时间的字段。

另一个指定的属性必须来自下表：

- **types** 为列表属性，该属性指定在列表中所允许的字段的日期类型，例如，**types="[range flag]"** 表示只列出上述存储类型的字段。可能的数据类型的设置如下：

范围

标志

set

orderedSet

numeric

discrete

typeless

- **onlyRanges**，如果设置为 **true**，则指定只列出数据类型为范围的字段。
- **onlyDiscrete** 如果设置为 **true**，则指定只列出数据类型为离散（即标志、设置或无类型）的字段。

因此，例如一个指定 **storage="[integer]"** 和 **types="[flag]"** 的控制确保只有类型为标志的整数字段会显示在列表中。

其余属性如下：

- **property** 是指用来存储字段值的属性的标识符。
- **multiple** 指定字段值是枚举列表 (**true**) 还是不是枚举列表 (**false**)。
- **label** 是控制的显示名称。
- **labelKey** 标识标签以便进行本地化。

输出字段

OutputFields 元素定义用户可以为模型选择一个或多个输出字段（即目标）的一组字段。

此设置组成了可在此节点中看到的所有字段。如果字段从此节点上游进一步过滤，那么只有通过过滤器的字段才是可见的。该列表可以进一步对之进行限制，其方式是指定仅可选择特定存储和数据类型的字段。

```
<OutputFields storage="storage_types" onlyNumeric="true_false" onlySymbolic="true_false"
  onlyDatetime="true_false" types="data_types" onlyRanges="true_false"
  onlyDiscrete="true_false" property="property_name" multiple="true_false" label="标签"
  labelKey="label_key"/>
```

通过指定两个属性（其中一个属性必须来自于下面的列表），可以将要使用的字段列表限制为输出字段：

- **storage** 是列表属性，该属性指定在列表中所允许的字段的存储类型，例如，**storage="[integer real]"** 表示只列出上述存储类型的字段。有关可能存储类型的设置的信息，请参阅数据和存储类型第 202 页码下的表格。
- **onlyNumeric** 如果设置为 **true**，则指定只列出存储类型为数字的字段。
- **onlySymbolic** 如果设置为 **true**，则指定只列出存储类型为符号（即字符串）的字段。
- **onlyDatetime** 如果设置为 **true**，则指定只列出存储类型为日期和时间的字段。

另一个指定的属性必须来自下表：

- **types** 为列表属性，该属性指定在列表中所允许的字段的日期类型，例如，**types="[range flag]"** 表示只列出上述存储类型的字段。可能的数据类型的设置如下：

范围

标志

set

orderedSet

numeric

discrete

typeless

- **onlyRanges**，如果设置为 **true**，则指定只列出数据类型为范围的字段。
- **onlyDiscrete** 如果设置为 **true**，则指定只列出数据类型为离散（即标志、设置或无类型）的字段。

因此，例如一个指定 **storage="[integer]"** 和 **types="[flag]"** 的控制确保只有类型为标志的整数字段会显示在列表中。

其余属性如下：

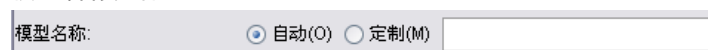
- **property** 是指用来存储字段值的属性的标识符。

- `multiple` 指定字段值是枚举列表 (`true`) 还是不是枚举列表 (`false`)。
- `label` 是控制的显示名称。
- `labelKey` 标识标签以便进行本地化。

模型生成

`ModelGeneration` 元素指定要在文件的其他位置所使用的标识符，从而定义模型构建器节点对话框的哪个选项卡将包含生成的模型的模型名称控件：

图片 5-2
模型名称控件



格式为：

```
<ModelGeneration controlId="control_identifier" />
```

`controlId` 属性指定随后在模型构建器节点规范的“用户界面”部分中的 `SystemControls` 元素中使用的标识符。其规范包含此 `SystemControls` 元素的选项卡将包含模型名称控件。

模型字段

`ModelFields` 元素可用于构造**模型签名** - 用于使用此模型评分数据的输入和输出字段组。

```
<ModelFields inputDirections="[in]" outputDirections="[out]">
  <AddField prefix="field_prefix" ... />
  ...
  <ForEach ...>
    <AddField prefix="field_prefix" ... />
  </ForEach>
  ...
</ModelFields>
```

其中 `inputDirections` 和 `outputDirections` 指定将如何构建模型签名；其值可以为 `in`、`out` 或 `both`。

这些字段本身由一个或多个 `AddField` 元素指定。`prefix` 属性指定要添加到字段名称中用来指示由模型所生成的字段的前缀。例如，如果字段名称是 `field1` 并且前缀值是 `$S`，则生成的字段名称为 `$S-field1`。[有关详细信息，请参阅第 72 页码第 4 章中的添加字段。](#)

通过 `ForEach` 元素可以进行迭代。[有关详细信息，请参阅第 75 页码第 4 章中的使用 ForEach 元素进行迭代。](#)

字段组允许对两个或两个以上的模型输出字段进行分组，以供迭代使用。输出字段名称后面附加后缀以指示迭代，例如 `$S-field1-1`、`$S-field1-2` 等等。使用一次字段组可以导致相同字段集在模型输出中多次出现。[有关详细信息，请参阅第 94 页码字段组示例。](#)

自动建模

AutoModeling 元素允许模型被整体建模节点使用，例如自动分类器、自动聚类或自动数值。有关详细信息，请参阅第 101 页码自动建模。

模型构建器示例

以下显示了交互节点示例的规范文件中完整的模型构建器部分（请参阅模型构建器节点 (Interaction) 第 32 页码）：

```
<Node id="interaction.builder" type="modelBuilder" palette="modeling" label="Interaction">
  <ModelBuilder miningFunctions="[classification]">
    <Algorithm value="robd" label="Robert's Algorithm" />
    <ModelingFields controlsId="modellingFields">
      <InputFields property="inputs" multiple="true" label="Inputs" onlyDiscrete="true" />
      <OutputFields property="target" multiple="false" label="Target" onlyDiscrete="true" />
    </ModelingFields>
    <ModelFields inputDirections="[in]" outputDirections="[out]">
      <ForEach var="field" inFields="outputs">
        <AddField prefix="$I" name="{field}" fieldRef="{field}" role="predictedValue"
          targetField="{field}" />
        <AddField prefix="$IP" name="{field}" storage="real" role="probability"
          targetField="{field}">
          <Range min="0.0" max="1.0"/>
        </AddField>
      </ForEach>
    </ModelFields>
  </ModelBuilder>
  ...
</Node>
```

字段组示例

本示例取自 SLRM 节点，它将由两个新字段组成的字段组添加到模型签名，以便在对模型评分时包含产生的数据。对于每条输入记录，为每个新字段进行用户指定次数的数据评分，评分次数由 `max_predictions` 属性值决定。

两个新字段为：

- `$S-target` - 包含目标字段的预测值
- `$SC-target` - 包含此预测的概率值

为了将这两个字段分组在一起，当在 `ModelFields` 部分中声明它们时为其指定相同的组标识符。组标识符通过 `AddField` 元素的 `group` 属性进行指定。

因此，在模型构建器节点的声明中包含下列内容：

```
<Node ... type="modelBuilder" ...>
  <ModelBuilder ...>
    ...
    <ModelFields inputDirections="[in]" outputDirections="[out]">
      <AddField prefix="$S" name="{target}" fieldRef="{target}" role="predictedValue"
        targetField="{target}" group="[1]" />
      <AddField prefix="$SC" name="{target}" storage="real" role="probability"
```

```

        targetField="${target}" group="[1]">
        <Range min="0.0" max="1.0"/>
    </AddField>
</ModelFields>
</ModelBuilder>
</Node>

```

因此，在模型填充器节点的声明中包含下列内容：

```

<Node ... type="modelApplier" ...>
...
    <OutputDataModel mode="extend">
        <ForEach var="group" from="1" to="${max_predictions}">
            <ForEach var="field" inFields="modelOutputs" container="model">
                <AddField name="${field}" group="[${group}]" fieldRef="${field}" />
            </ForEach>
        </ForEach>
    </OutputDataModel>
</Node>

```

目标字段命名为 `campaign`，用户在对应于 `max_predictions` 属性的字段中输入 2。执行模型构建器节点将导致以下字段被附加到模型：

- `$$-campaign-1`
- `$$C-campaign-1`
- `$$-campaign-2`
- `$$C-campaign-2`

模型输出

`ModelOutput` 元素所描述的模型输出对象将在执行流后，显示在管理器窗格的“模型”选项卡下。

格式

```

<ModelOutput id="identifier" label="display_label" labelKey="label_key" >
    <ModelProvider .../>
    <属性>
        <Property .../>
        ...
    </Properties>
    <Containers .../>
    <UserInterface .../>
    <Constructors .../>
</ModelOutput>

```

其中：

- `id`（必需）是生成的模型的唯一标识符。
- `label`（必需）是所生成的模型将显示在“模型”选项卡上的显示名称。
- `labelKey` 标识标签以便进行本地化。

可以包含在 `ModelOutput` 元素中的子元素如下所示：

表 5-3
模型输出声明的子元素

子元素	定义	请参阅...
<code>ModelProvider</code>	控制模型输出（无论是否为 PMML 格式）的容器。	模型提供器第 57 页码
属性	生成的模型要使用的属性。	Properties第 58 页码
<code>Containers</code>	放置生成的模型输出的容器。	Containers第 59 页码
<code>User Interface</code>	可以从中查看生成的模型输出的用户界面，例如，模型输出对象上的“模型”和“设置”选项卡。	用户界面第 61 页码
<code>Constructors</code>	生成的模型产生的对象。	使用构造函数第 109 页码

示例

```
<ModelOutput id="interaction.model" label="Interaction Model">
  <Properties>
  </Properties>
  <Containers>
    <Container name="model" />
  </Containers>
  <User Interface>
    <Tabs>
      <Tab label="Model">
        <TextBrowserPanel container="model" textFormat="plainText" />
      </Tab>
    </Tabs>
  </User Interface>
  <Constructors>
    <CreateModelApplier type="interaction.applier">
      <SetContainer target="model" source="model" />
    </CreateModelApplier>
  </Constructors>
</ModelOutput>
```

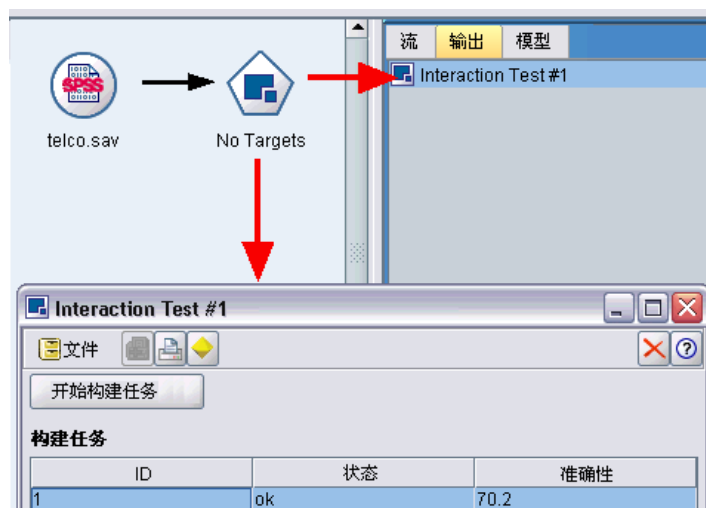
构建交互模型

通过交互建模功能，您可以创建最终用户在生成模型之前可用于交互的输出对象。此交互输出对象位于管理器窗格的“输出”选项卡上，并且包含一个临时数据集。临时数据集可用于在模型生成之前对其进行精练或简化。通过将一些额外元素添加到常规模型构建器节点的规范中可实现交互建模：

- `Node` 定义的 `Constructors` 部分包含 `CreateInteractiveModelBuilder` 元素。
- 扩展包括专用的 `InteractiveModelBuilder` 元素。

通过名为**交互窗口**的窗口（创建输出对象后即可显示该窗口），用户可以与临时数据集进行交互。

图片 5-3
交互输出对象和交互窗口



交互操作特定于所使用的算法，可通过扩展实现。交互窗口是在 `InteractiveModelBuilder` 元素的“用户界面”部分定义的。可以通过指定以下任一内容定义交互窗口：

- 帧类（请参阅[用户界面部分第 116 页码](#)），它完整定义了窗口
- 面板类，将其指定为窗口每个选项卡的扩展对象面板的属性（请参阅[扩展对象面板第 129 页码](#)）

关闭后，通过在“输出”选项卡上双击对象名称可以重新显示交互窗口。

交互窗口规范需要包含节点才能在用户完成交互后生成模型。在所说明的示例中，此操作是通过带有金色块图标的工具栏按钮实现的，该按钮与生成模型的操作相关。此操作的代码显示在[交互建模示例第 99 页码](#)下面的 `InteractiveModelBuilder` 部分。

创建交互模型构建器

`CreateInteractiveModelBuilder` 元素描述了用户用于交互的输出对象。它是 `CreateModelOutput` 元素的一个有效交互版本。

格式

此元素可用于模型构建器节点定义的“执行”部分：

```
<Node ... type="modelBuilder" ...>
...
  <Execution>
    ...
      <Constructors>
        ...
          <CreateInteractiveModelBuilder ...>
            ...
          </CreateInteractiveModelBuilder>
```

```

        </Constructors>
    </Execution>
    ...
</Node>

```

元素本身的格式为：

```

<CreateInteractiveModelBuilder type="output_object_id">
    <Condition .... />
    <And>
    <Or>
    <Not>
    <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
    <CreateDocument type="model_id" target="container_id" sourceFile="container_file_id" />
</CreateInteractiveModelBuilder>

```

其中 **type**（必需）是由 **InteractiveModelBuilder** 元素创建的输出对象的标识符。

Condition 部分允许指定一个或多个条件。 [有关详细信息，请参阅第 80 页码第 4 章中的条件。](#)

还可以指定包含 **And**、**Or** 以及 **Not** 运算符的复杂条件。 [有关详细信息，请参阅第 79 页码第 4 章中的逻辑运算符。](#)

在 **CreateModel** 元素和 **CreateDocument** 元素中：

- **type**是正在定义的模型或文档的标识符。
- **target**（必需）是模型中容器的标识符；在“模型输出”部分对此类容器进行了定义。 [有关详细信息，请参阅第 95 页码模型输出。](#)
- **sourceFile**（必需）是节点执行期间生成的输出文件的标识符；在“输出文件”部分对此类输出文件进行了定义。 [有关详细信息，请参阅第 63 页码第 4 章中的输出文件。](#)

示例

```

<CreateInteractiveModelBuilder type="my.interaction">
    <Condition property="interactive" op="equals" value="true" />
</CreateInteractiveModelBuilder>

```

此示例指定在执行规范文件中包含此元素的模型构建器节点时，会创建带有 **my.interaction** 标识符的输出对象。输出对象本身由引用此标识符的 **InteractiveModelBuilder** 元素在规范文件的其他地方定义，例如：

```

<InteractiveModelBuilder id="my.interaction" label=...>
    ...
</InteractiveModelBuilder>

```

交互模型构建器

此元素可定义交互输出对象，该对象允许最终用户在模型生成之前对其进行精练或简化。

InteractiveModelBuilder 元素符合包含相应 **CreateInteractiveModelBuilder** 元素的模型构建器节点的定义。 [有关详细信息，请参阅第 97 页码创建交互模型构建器。](#)

格式

`InteractiveModelBuilder` 元素的格式为：

```
<Node ... type="modelBuilder" ...>
  ...
  -- Create Interactive Model Builder section --
  ...
</Node>
...
<InteractiveModelBuilder id="identifier" label="display_label" labelKey="label_key">
  <Properties>
    <Property name=.../>
    ...
  </Properties>
  <Containers>
    <Container name="container_name"/>
  </Containers>
  <UserInterface .../>
  <Constructors .../>
</InteractiveModelBuilder>
```

其中：

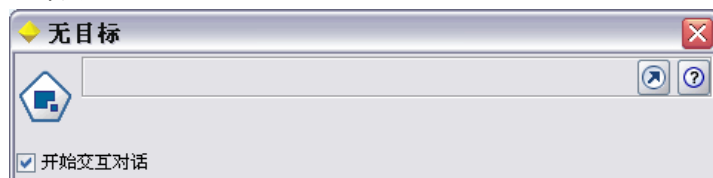
- `id`（必需）是生成的模型的唯一标识符。
- `label`（必需）是所生成的模型将显示在“模型”选项卡上的显示名称。
- `labelKey` 标识标签以便进行本地化。

有关 `Properties`、`Containers`、`UserInterface` 和 `Constructors` 元素的详细信息，请参阅[Properties](#)第 58 页码、[Containers](#)第 59 页码、[用户界面部分](#)第 116 页码和[使用构造函数](#)第 109 页码。

交互建模示例

本示例阐释如何通过以下方法定义模型构建器节点：用户可以通过一个简单的复选框选择是否在生成模型之前进行交互，例如：

图片 5-4
选择交互



要查看实际操作，请使用此版本所提供的交互示例节点。有关详细信息，请参阅第 32 页码第 3 章中的模型构建器节点（[Interaction](#)）。

首先，模型构建器节点指定布尔值属性：

```

<Node id="interaction.builder" type="modelBuilder" ...>
  ...
  <Properties>
    <Property name="交互式" valueType="boolean" />
  </Properties>

```

在节点规范的“用户界面”部分中，定义“模型”选项卡的部分包含对此属性的引用：

```

<Tab label="Model">
  <PropertiesPanel>
    <CheckBoxControl property="交互式" label="Start an interactive session" />
  </PropertiesPanel>
</Tab>

```

在同一节点的 `CreateInteractiveModelBuilder` 部分，将测试属性的设置，如果结果为 `true`，则将创建交互输出对象：

```

<CreateInteractiveModelBuilder type="my.interaction">
  <Condition property="交互式" op="equals" value="true" />
</CreateInteractiveModelBuilder>

```

它引用的输出对象在扩展的 `InteractiveModelBuilder` 部分中定义：

```

<InteractiveModelBuilder id="my.interaction" label="Interaction Test">
  <Properties>
  </Properties>
  <Containers>
  </Containers>
  <UserInterface actionHandler="ui.InteractionHandler">
    <Controls>
      <ToolBarItem action="generateModel" showLabel="false" />
    </Controls>
    <Tabs>
      <Tab label="Model">
        <ExtensionObjectPanel id="model.panel" panelClass="ui.SampleInteractionPanel" />
      </Tab>
      <Tab label="Generic">
        <ExtensionObjectPanel id="generic.panel" panelClass="ui.GenericInteractionPanel" />
      </Tab>
    </Tabs>
  </UserInterface>
</InteractiveModelBuilder>

```

生成模型的操作是由 `ToolBarItem` 元素定义的工具栏按钮控制的。

请注意，使用 `ExtensionObjectPanel` 元素的 `panelClass` 属性来指定 Java 类以控制交互窗口每个选项卡的用户界面。

自动建模

IBM® SPSS® Modeler 提供了一组标准整体建模节点，如自动分类器节点、自动聚类节点和自动数值节点。这些节点可自动同时构建多个不同的模型，从而允许最终用户对结果进行比较并为他们的数据选择最佳模型。CLEF 提供了 **AutoModeling** 元素可允许任意这些整体节点使用由 **ModelBuilder** 元素指定的模型。

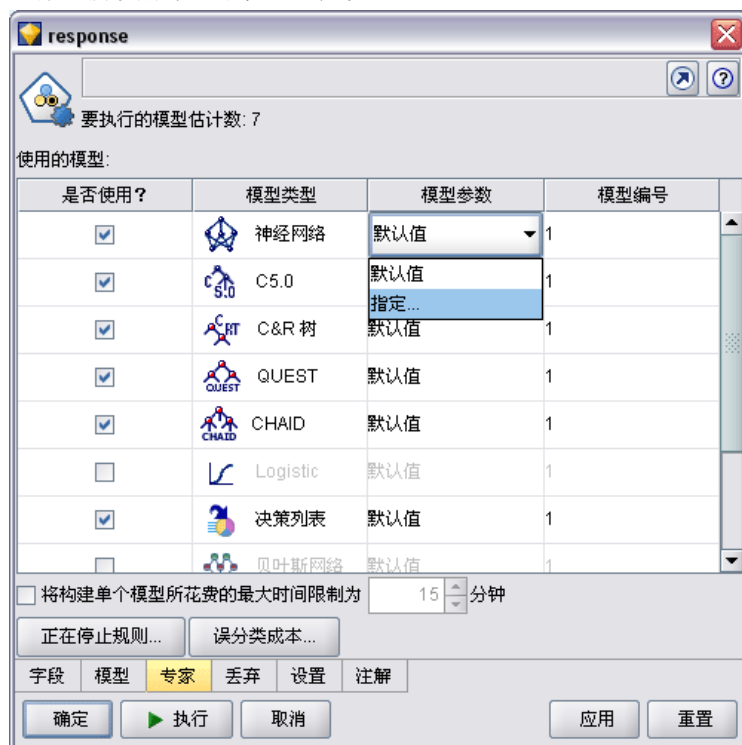
AutoModeling 元素的格式为：

```
<AutoModeling enabledByDefault="true_false">
  <SimpleSettings .../>
  <ExpertSettings .../>
  <Constraint .../>
  <Constraint .../>
  ...
</AutoModeling>
```

其中 **enabledByDefault** 指定是否在整体建模节点中默认启用模型（即，默认情况下为特定模型选中使用？列，如下图所示）。如果忽略该属性，则假定值为**真**。

整体建模节点对话框的“专家”选项卡中列出了用户可以选择构建的模型，例如：

图片 5-5
整体建模节点的“专家”选项卡



单击特定模型的模型参数字段中的指定可以显示“算法设置”对话框，用户可以在其中为该模型类型选择选项。

图片 5-6
整体建模节点的算法设置



“算法设置”对话框包含“简单”选项卡和“专家”选项卡，它们分别对应于建模节点的“简单”执行模式和“专家”执行模式。“简单”选项卡和“专家”选项卡的内容由 `SimpleSettings` 和 `ExpertSettings` 元素控制，这些元素将在下面几节中进行介绍。

此外，通过 `Constraint` 元素，您可以指定允许最终用户编辑或约束（在某些方面）“算法设置”对话框中的参数的条件。[有关详细信息，请参阅第 105 页码约束。](#)

“算法设置”对话框中的某些参数可以有多个值。在指定多个值的地方，整体节点将尝试为所有可能的参数值组合构建模型。例如，对于广义线性模型，如果用户指定两个分布（正态和伽马）和三个链接功能（恒等式、对数和幂），自动数值节点将尝试构建六个广义线性模型，分别对应于每一个可能的参数组合。

简单设置

对于整体建模节点中的这种模型，`SimpleSettings` 元素可确定“算法设置”对话框的“简单”选项卡上将显示哪些参数。[有关详细信息，请参阅第 101 页码自动建模。](#)

格式

```
<SimpleSettings>
  <PropertyGroup label="group_name" labelKey="resource_key" properties="[prop_name1
    prop_name2 ...]"/>
  <PropertyGroup .../>
</SimpleSettings>
```

在 `PropertyGroup` 元素中（必须至少有一个）：

`label` 是属性组的显示标签，它作为副标题插入到该属性组第一个参数前的对话框中。

`labelKey` 标识标签以便进行本地化。如果既没有使用 `label`，也没有使用 `labelKey`，则不插入副标题。

`properties`（必需）是要在选项卡中显示的一个或多个属性的列表。`prop_name1` 和 `prop_name2` 等的值是 `Property` 元素的 `name` 属性的值，该属性是该元素中定义的。[有关详细信息，请参阅第 58 页码第 4 章中的 Properties。](#)

示例

```
<SimpleSettings>
  <PropertyGroup properties="[method]" />
</SimpleSettings>
```

本示例来自“判别式”节点，指定只有 `Method` 参数将显示在相关整体建模节点模型的“算法设置”对话框的“简单”选项卡中（在此示例中，是自动分类器节点）。由于没有指定 `label` 或 `labelKey` 属性，因此没有在对话框中显示参数的子标题。

专家设置

对于整体建模节点中的这种模型，`ExpertSettings` 元素可确定“算法设置”对话框的“专家”选项卡上将显示哪些参数。[有关详细信息，请参阅第 101 页码自动建模。](#)

格式

```
<ExpertSettings>
  <Condition ... />
  <PropertyGroup label="group_name" labelKey="resource_key"
    properties="[property1 property2 ...]" />
  <PropertyGroup ... />
  ...
</ExpertSettings>
```

`Condition` 元素指定这样一个条件：如果该条件为 `true`，将导致由后续的一个或多个 `PropertyGroup` 元素标识的参数被启用。[有关详细信息，请参阅第 80 页码第 4 章中的条件。](#)

在 `PropertyGroup` 元素中（必须至少有一个）：

`label` 是属性组的显示标签，它作为副标题插入到该属性组第一个参数前的对话框中。

`labelKey` 标识标签以便进行本地化。如果既没有使用 `label`，也没有使用 `labelKey`，则不插入副标题。

`properties`（必需）是要在选项卡中显示的一个或多个属性的列表。`prop_name1` 和 `prop_name2` 等的值是 `Property` 元素的 `name` 属性的值，该属性是该元素中定义的。[有关详细信息，请参阅第 58 页码第 4 章中的 Properties。](#)

示例

在下面的示例中，“算法设置”对话框的“专家”选项卡将模式参数最初设置为简单：

图片 5-7
专家设置被禁用



下面的内容指定只有当用户将模式参数设置更改为专家时，才可以启用其他“专家”选项卡参数：

```
<ExpertSettings>
  <Condition property="mode" op="equals" value="Expert"/>
  <PropertyGroup properties="[mode prior_probabilities covariance_matrix]"/>
  ...
</ExpertSettings>
```

将模式设置更改为专家则会在属性组中启用这两个参数：

图片 5-8
专家设置被启用



下一个示例阐释属性组标签的用法：

```
<ExpertSettings>
  ...
  <PropertyGroup labelKey="automodel_stepping_criteria_options"
    properties="[stepwise_method V_to_enter criteria]"/>
  ...
</ExpertSettings>
```

这里，PropertyGroup 元素控制示例中突出显示的参数：

图片 5-9
专家设置被禁用



labelKey 属性允许 CLEF 检索扩展的属性文件中对应条目的属性组子标题的显示文本:

automodel.stepping_criteria_options=Stepping Criteria option(s)

有关详细信息, 请参阅第 186 页码第 8 章中的属性文件。

约束

Constraint 元素指定允许编辑的条件, 或在某些方面对在整体建模节点中某个模型的“算法设置”对话框中列出的参数的约束条件。例如, 如果当前不允许最终用户修改某些参数, 则这些参数可能被禁用。

格式

```
<Constraint property="prop_name" singleSelection="true_false">
  <Condition property="prop_name" op="operator" value="值"/>
  <And .../>
  <Or .../>
  <Not .../>
</Constraint>
```

其中:

- **property**（必需）标识要编辑或约束的参数。`prop_name` 是定义对应于此参数的属性中的 **Property** 元素的 `name` 属性的值。有关详细信息，请参阅第 58 页码第 4 章中的 **Properties**。
- **singleSelection** 控制最终用户是否可以为用户选择多个可用值。如果设置为 `true`，则即使在“算法设置”对话框的该参数的“选项”字段中列出了多个值，也只能选择一个值。如果设置为 `false`（默认），最终用户可以选择一个或多个可用值，如下面的示例所示。

Condition 元素指定实际的约束。有关详细信息，请参阅第 80 页码第 4 章中的条件。**And**、**Or** 和 **Not** 元素可用于指定复合条件。有关详细信息，请参阅第 79 页码第 4 章中的逻辑运算符。

示例

本示例取自于广义线性节点的规范文件。默认情况下，即使在“专家”模式中，某些参数也无法启用。

图片 5-10
约束的效果 - 参数被禁用



约束指定在其下启用值参数的条件：

```
<Constraint property="scale_value">
  <And>
    <Condition property="scale_method" op="equals" value="FixedValue"/>
    <Condition property="distribution" op="in" value="[1GAUSS GAMMA NORMAL]"/>
  </And>
</Constraint>
```

尺度参数方法参数（由 `scale_method` 属性标识）必须设置为固定值，并且分布必须在启用值参数之前设置为正态、逆高斯或伽玛。

图片 5-11
约束的效果 - 参数被启用



应用模型

应用模型是指使用生成的模型对数据进行评分，也就是说，使用从模型构建中获得的信息为新记录创建预测值。在 IBM® SPSS® Modeler 中，您可以通过模型填充器节点实现此操作。有关详细信息，请参阅第 13 页码第 2 章中的模型填充器节点。

在规范文件中定义模型填充器节点会创建框架以便用于应用生成的模型。在 SPSS Modeler 中，通过将表示管理器窗格“模型”选项卡的模型输出对象的图标拖放到流工作区上，可创建模型填充器节点的实例。如果没有模型填充器节点定义，执行模型构建器节点将只生成非精练模型，该模型无法添加到流工作区中。

定义模型填充器节点时，Node 元素必须包括：

- type="modelApplier" 属性
- 包含 CreateModelOutput 元素的 Constructors 子元素（请参阅使用构造函数第 109 页码）

有关 Node 元素规范的格式信息，请参阅节点第 54 页码。

构建文档

定义文档构建器节点时，Node 元素必须包含：

- type="documentBuilder" 属性
- DocumentBuilder 子元素

构建文档时，扩展还需要 DocumentOutput 元素来描述生成的文档。有关详细信息，请参阅第 95 页码模型输出。

有关 Node 元素规范的格式信息，请参阅节点第 54 页码。

文档构建器

`DocumentBuilder` 元素定义文档构建器节点的行为。定义必须包含 `DocumentGeneration` 子元素以便指定文档构建器节点对话框中的哪个选项卡将包含文档生成控件。控件本身是在“用户界面”部分定义的（请参阅第 6 章、构建用户界面 第 113 页码）。

格式

```
<DocumentBuilder>
  <DocumentGeneration controlId="control_identifier" />
</DocumentBuilder>
```

其中 `controlId`（必需）是系统控件用于指定文档生成控件出现位置的标识符。

示例

```
<DocumentBuilder>
  <DocumentGeneration controlId="1" />
</DocumentBuilder>
```

文档输出

`DocumentOutput` 元素所描述的文档输出对象将在执行流后，显示在管理器窗格的“输出”选项卡下。

格式

```
<DocumentOutput id="identifier" label="display_label" labelKey="label_key" >
  <Properties>
    <Property .../>
    ...
  </Properties>
  <Containers>
    <Container .../>
    ...
  </Containers>
  <UserInterface .../>
  <Constructors .../>
</DocumentOutput>
```

其中：

- `id`（必需）是生成的文档的唯一标识符。
- `label`（必需）是所生成的文档将显示在“输出”选项卡上的显示名称。
- `labelKey` 标识标签以便进行本地化。

可以包含在 `DocumentOutput` 元素中的子元素如下所示：

表 5-4
文档输出声明的子元素

子元素	定义	请参阅...
属性	生成的文档要使用的属性。	Properties 第 58 页码
Containers	将放置生成的文档输出的容器。	Containers 第 59 页码
User Interface	通过其可以查看生成的文档输出的用户界面。	用户界面 第 61 页码
Constructors	生成的文档产生的对象。	使用构造函数 第 109 页码

示例

```
<DocumentOutput id="webstatusreport">
  <Containers>
    <Container name="webstatusreportdata" />
  </Containers>
  <User Interface>
    <Tabs>
      <Tab label="Advanced" labelKey="advancedTab.LABEL">
        <TextBrowserPanel container="webstatusreportdata" textFormat="html" />
      </Tab>
    </Tabs>
  </User Interface>
</DocumentOutput>
```

使用构造函数

`Constructors` 元素可以显示在规范文件中以下两个位置中的任意一处：

- 在节点定义的“执行”部分（针对模型或文档输出对象）
- 在模型输出定义中（针对模型填充器节点）

一个节点只能生成一个输出对象。施加此限制是为了保证与现有节点、脚本编写以及这些类型节点的客户端 API 接口保持一致。。

格式

在“执行”部分使用时，`Constructors` 元素的格式为：

```
<Constructors>
  <CreateModelOutput .../>
  ...
  <CreateDocumentOutput .../>
  ...
  <CreateInteractiveModelBuilder .../>
  ...
</Constructors>
```

在模型输出定义中，格式为：

```

<Constructors>
  <CreateModelApplier .../>
</Constructors>

```

创建模型输出

此部分定义如何在“模型”选项卡上创建模型输出对象，或如何在“输出”选项卡上创建文档输出对象。

格式

```

<CreateModelOutput type="output_object_id">
  <Condition ..../>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelOutput>

```

在 `CreateModelOutput` 元素中：

- `type`（必需）是在“模型输出”部分定义的模型输出对象的标识符。有关详细信息，请参阅第 56 页码第 4 章中的模型输出。

`Condition` 部分允许指定一个或多个条件。有关详细信息，请参阅第 80 页码第 4 章中的条件。

还可以指定包含 `And`、`Or` 和 `Not` 运算符的复杂条件。有关详细信息，请参阅第 79 页码第 4 章中的逻辑运算符。

在 `CreateModel` 元素和 `CreateDocument` 元素中：

- `type` 是正在定义的模型或文档的标识符。
- `target`（必需）是模型中容器的标识符；在“模型输出”部分对此类容器进行了定义。有关详细信息，请参阅第 95 页码模型输出。
- `sourceFile`（必需）是节点执行期间生成的输出文件的标识符；在“输出文件”部分对此类输出文件进行了定义。有关详细信息，请参阅第 63 页码第 4 章中的输出文件。

示例

```

<Constructors>
  <CreateModelOutput type="naivebayes">
    <CreateModel type="naivebayes_model" target="model" sourceFile="pmm1"/>
    <CreateDocument type="html_output" target="advanced_output" sourceFile="htmloutput" />
    <CreateDocument type="zip_outputType" target="zip_output" sourceFile="zipoutput" />
  </CreateModelOutput>
</Constructors>

```

创建文档输出

此元素可在文档构建器节点定义的“执行”部分使用，并且可以标识正在创建的文档输出对象。

格式

```
<CreateDocumentOutput type="output_object_id">
  <Condition ..../>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelOutput>
```

其中 **type**（必需）是在“文档输出”部分定义的文档输出对象的标识符。[有关详细信息，请参阅第 57 页码第 4 章中的文档输出。](#)

Condition 部分允许指定一个或多个条件。[有关详细信息，请参阅第 80 页码第 4 章中的条件。](#)

还可以指定包含 **And**、**Or** 和 **Not** 运算符的复杂条件。[有关详细信息，请参阅第 79 页码第 4 章中的逻辑运算符。](#)

在 **CreateModel** 元素和 **CreateDocument** 元素中：

- **type**是正在定义的模型或文档的标识符。
- **target**（必需）是模型中容器的标识符；在“模型输出”部分对此类容器进行了定义。[有关详细信息，请参阅第 95 页码模型输出。](#)
- **sourceFile**（必需）是节点执行期间生成的输出文件的标识符；在“输出文件”部分对此类输出文件进行了定义。[有关详细信息，请参阅第 63 页码第 4 章中的输出文件。](#)

示例

```
<Constructors>
  <CreateDocumentOutput type="webstatusreport">
    <CreateDocument type="webstatusreport" target="webstatusreportdata"
      sourceFile="webstatusreport_output_file" />
  </CreateDocumentOutput>
</Constructors>
```

创建交互模型构建器

此元素可在交互模型构建器节点定义的“执行”部分中使用，并且可以标识用户用于交互的输出对象。[有关详细信息，请参阅第 97 页码创建交互模型构建器。](#)

创建模型填充器

此元素在模型构建器节点定义的“模型输出”部分中的 `Constructors` 元素内使用（请参阅[模型输出第 95 页码](#)）。`CreateModelApplier` 元素定义在将模型构建器节点生成的模型输出对象拖放到工作区时如何创建模型填充器节点。

格式

```
<CreateModelApplier type="apply_node_identifier">
  <Condition .../>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelApplier>
```

在 `CreateModelApplier` 元素中：

- `type`（必需）是要创建的模型填充器节点的标识符；实际上，随后将在文件的 `Node ... type="modelApplier"` 元素中定义该节点。

`Condition` 部分允许指定一个或多个条件。 [有关详细信息，请参阅第 80 页码第 4 章中的条件。](#)

还可以指定包含 `And`、`Or` 以及 `Not` 运算符的复杂条件。 [有关详细信息，请参阅第 79 页码第 4 章中的逻辑运算符。](#)

在 `CreateModel` 元素和 `CreateDocument` 元素中：

- `type`是正在定义的模型或文档的标识符。
- `target`（必需）是模型中容器的标识符；在“模型输出”部分对此类容器进行了定义。 [有关详细信息，请参阅第 95 页码模型输出。](#)
- `sourceFile`（必需）是节点执行期间生成的输出文件的标识符；在“输出文件”部分对此类输出文件进行了定义。 [有关详细信息，请参阅第 63 页码第 4 章中的输出文件。](#)

示例

在下面的示例中，`CreateModelApplier` 元素包含一个对名为 `myapplier` 的模型填充器节点的前向引用。该节点在随后的 `Node` 元素中定义。

```
<ModelOutput>
  <Constructors>
    <CreateModelApplier type="myapplier"></CreateModelApplier>
  </Constructors>
</ModelOutput>
<Node id="myapplier" type="modelApplier">
  ...
</Node>
```

构建用户界面

关于用户界面

当添加新的 CLEF 节点时，需要定义最终用户与节点、任何模型输出和文档输出进行交互的方式，或者定义如何应用扩展所指定的节点。每个对象的用户界面是在扩展的规范文件的 **UserInterface** 部分进行定义的。本章将详细介绍 **UserInterface** 部分的内容。

注：根据文件中所定义的对象类型，单个规范文件中可能存在多个 **UserInterface** 部分。

CLEF 对象的用户界面由以下一项或多项组成：

- 菜单项
- 选项板项
- Icons
- 一个或多个对话框
- 一个或多个输出窗口

通过**菜单项**和**选项板项**，用户可以从 IBM® SPSS® Modeler 菜单系统和节点选项板分别访问对象。**图标**可识别菜单、画布以及多种节点选项板中的对象。**对话框**包含选项卡和控件，允许用户在执行流之前指定各种选项，还可以在执行完成后指定要生成的输出。**输出窗口**用于显示模型输出（如，对数据集应用模型时生成的结果）或文档输出（如图表）。

CLEF 允许您将新的对象类型添加到由 SPSS Modeler 提供的标准类型中，并且还支持使用一致的方法为这些新对象定义用户界面。有关在 CLEF 中创建图标和对话框的指导准则，请参阅[设计节点图标第 16 页码](#)和[设计对话框第 20 页码](#)。

图标采用了图形的格式，这些图形可标识特殊节点，并且显示在标识节点类型的几何形状内部。

对话框和**输出窗口**具有以下特征：

- 包含缩图图标和对象标题的标题栏
- 菜单栏内容包括：
 - 菜单
 - 特定于对象的操作按钮
 - 常用操作按钮（例如，最大化或帮助）
- 主要内容区域
- 用于将 UI 组件组织到逻辑组中的多个选项卡
- 重新调整容量大小

选项卡可以以多种不同的方式更改窗口的主要内容区域。对于对话框，不同的选项卡显示不同对象属性的控件集。可以对这些控件进行修改，并且当用户单击应用或确定按钮时，修改的结果将应用到基础对象中。

对于输出窗口，用户使用选项卡可以执行与输出相关的不同操作，例如显示结果汇总或将对结果添加注解。

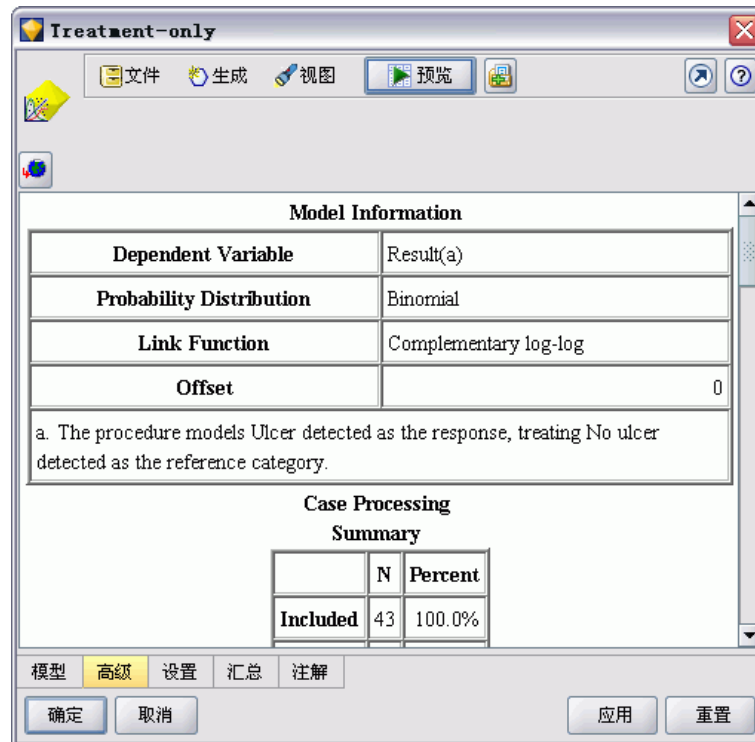
以下示例显示了具有多种选项卡和控件的简单对话框：

图片 6-1
选项卡对话框



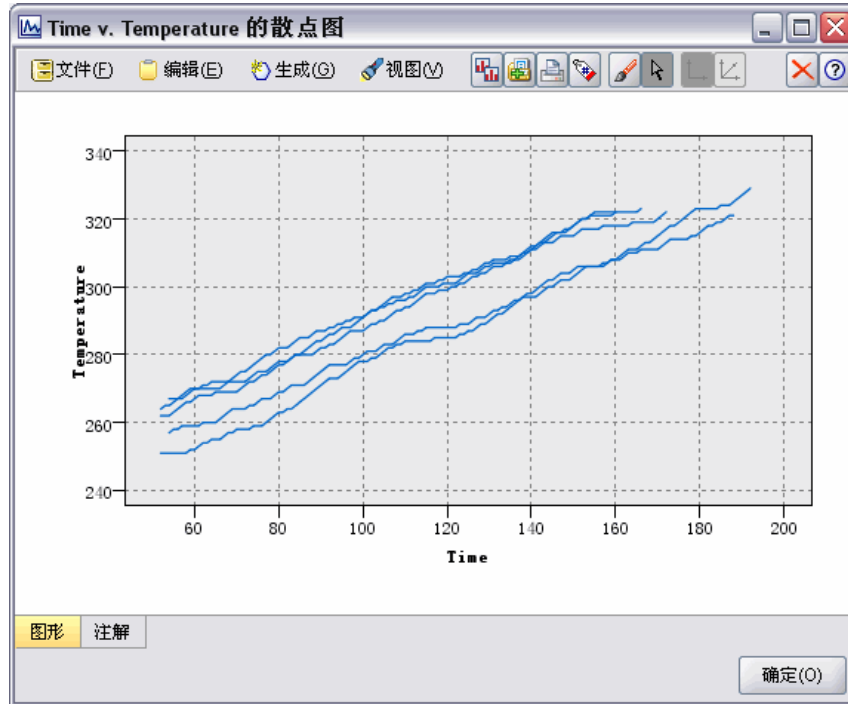
以下示例显示了模型输出窗口：

图片 6-2
模型输出窗口



最后为文档输出窗口示例（在本示例中显示为图表）：

图片 6-3
文档输出窗口



用户界面部分

对象的用户界面是在规范文件中对象定义内的 **UserInterface** 元素部分进行声明的。在同一规范文件中，可以有一个或多个 **UserInterface** 元素，这取决于文件中所定义对象（例如，模型构建器节点、模型输出对象和模型填充器节点）的数量。

在每个用户界面部分中，您都可以定义以下内容：

- 在工作区或选项板中显示的图标
- 对话框或输出窗口中显示的控件（自定义菜单和工具栏项）
- 定义属性控件集的选项卡（针对于对话框或输出窗口）

注：在下面的元素定义（通常由标题**格式**标识）中，除非指示为“（必需）”，否则元素属性和子元素均为可选项。有关所有元素的完整语法信息，请参阅附录 A，CLEF XML 架构第 224 页码。

对于每一个用户界面，可以指定要执行的处理操作。您可以通过操作处理程序或帧类属性执行此项操作，这两种方法都可以选择。如果操作处理程序或帧类属性都未作指定，将在文件的其他位置指定要执行的处理操作。

格式

`User Interface` 元素的基本格式为：

```
<User Interface>
  <Icons>
    <Icon .../>
    ...
  </Icons>
  <Controls>
    <Menu .../>
    <MenuItem .../>
    ...
    <ToolBarItem .../>
    ...
  </Controls>
  <Tabs>
    <Tab .../>
    ...
  </Tabs>
</User Interface>
```

当您自定义操作添加至标准 IBM® SPSS® Modeler 窗口时可使用**操作处理程序**。操作处理程序可以指定 Java 类，这些 Java 类将在用户选择节点对话框、模型输出窗口或文档输出窗口中的自定义菜单选项或工具栏按钮时被调用。它可以实现 `ExtensionObjectFrame` 类或 `ActionHandler` 类。无论在何种情况下，都将自动包含标准窗口组件，如标准菜单、选项卡以及工具栏按钮。[有关详细信息，请参阅第 193 页码第 9 章中的客户端 API 类。](#)

除第一行以外，操作处理程序的格式与基本格式均相同：

```
<User Interface actionHandler="Java_class" >
  ...
</User Interface>
```

其中 `actionHandler` 是操作处理程序 Java 类的名称。

帧类可用于模型输出或文档输出对象，其中扩展将提供其自身的窗口而无需自定义标准 SPSS Modeler 窗口。帧类是一种可完全指定整个窗口及其处理的 Java 类。不会自动包含标准窗口组件 - 类必须逐个指定这些组件。帧类只能用于模型输出或文档输出对象，而不能用于节点（其经常使用 SPSS Modeler 对话框）。[有关详细信息，请参阅第 178 页码自定义输出窗口。](#)

帧类的格式比较简单：

```
<UserInterface frameClass="Java_class" />
```

其中 `frameClass` 是模型输出或文档输出对象的帧类名称。任何图标、控件和选项卡都由帧类自身指定，因此不以此格式使用那些元素。

`User Interface` 元素的子元素将在随后的章节中进行介绍。

示例

第一个示例显示模型构建器节点的用户界面：

```
<UserInterface actionHandler="com.spss.myextension.MyActionHandler">
  <Icons>
    <Icon type="standardNode" imagePath="images/lg_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_discriminant.gif" />
  </Icons>
  <Tabs defaultTab="1">
    ...
  </Tabs>
</UserInterface>
```

对应的模型输出对象部分为：

```
<UserInterface>
  <Icons>
    <Icon type="standardWindow" imagePath="images/browser_discriminant.gif" />
  </Icons>
  <Tabs>
    <Tab label="Advanced" labelKey="advancedTab.LABEL"
      helpLink="discriminant_output_advancedtab.htm">
      <ExtensionObjectPanel id="DiscriminantPanel"
        panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
    </Tab>
  </Tabs>
</UserInterface>
```

模型填充器节点的用户界面部分如下所示：

```
<UserInterface>
  <Icons>
    <Icon type="standardNode" imagePath="images/lg_gm_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_gm_discriminant.gif" />
  </Icons>
  <Tabs>
    <Tab label="Advanced" labelKey="advancedTab.LABEL"
      helpLink="discriminant_output_advancedtab.htm">
      <ExtensionObjectPanel id="DiscriminantPanel"
        panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
    </Tab>
  </Tabs>
</UserInterface>
```

Icons

本部分定义与对象相关的图标。

对于节点，本部分应定义以下两个 `Icon` 元素：

- 在工作区中显示的较大图标
- 在选项板上显示的较小图标

对于模型输出和文档输出，这将定义显示在窗口框架左上角的缩图图标。

有关在 CLEF 中创建图标的指导准则，请参阅[设计节点图标第 16 页码](#)。




格式

```
<Icons>
  <Icon type="icon_type" imagePath="image_path" />
  ...
</Icons>
```

其中：

type（必需）是以下图标类型中的一种：

表 6-1
图标类型

类型	示例	说明
standardNode		工作区中以节点形状显示的大尺寸（49 x 49 像素）图标。
smallNode		节点选项板上以节点形状显示的小尺寸（38 x 38 像素）图标。
窗口		浏览器或输出窗口中显示的缩图（16 x 16 像素）图标。

imagePath（必需）可标识图标中所用图像的位置。该位置根据规范文件安装所在的目录进行指定。

示例

```
<Icon type="standardNode" imagePath="images/lg_mynode.gif" />
<Icon type="smallNode" imagePath="images/sm_mynode.gif" />
<Icon type="window" imagePath="images/mynode16.gif" />
```

Controls

本部分定义自定义菜单和工具栏项，这些项可用于调用在规范文件的“通用对象”部分声明的操作。有关详细信息，请参阅[第 44 页码第 4 章中的 Actions](#)。

格式

```
<Controls>
  <Menu .../>
  ...
```

```

    <MenuItem .../>
    ...
    <ToolBarItem .../>
    ...
</Controls>

```

`Menu`、`MenuItem` 和 `ToolBarItem` 元素将在随后的章节中进行介绍。

示例

以下示例将一个项添加到“生成”菜单以及对话框（该对话框的规范中包含该项）的工具栏中。两个项目都将完成先前定义的名为 `generateDerive`（生成“导出”节点）的操作。

```

<Controls>
  <MenuItem action="generateDerive" systemMenu="generate"/>
  <ToolBarItem action="generateDerive" showLabel="false"/>
  ...
</Controls>

```

菜单

可以定义自定义菜单以添加到其中一个标准菜单中。

格式

```

<Menu id="名称" label="display_label" labelKey="label_key" systemMenu="menu_name"
  showLabel="true_false" showIcon="true_false" separatorBefore="true_false" separatorAfter="true_false"
  offset="整数" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"/>

```

其中：

`id`（必需）是要添加的菜单的唯一标识符。

`label`（必需）是菜单出现在用户界面上的显示名称（前提是将 `showLabel` 设置为 `true`）。

`labelKey` 标识标签以便进行本地化。

`systemMenu`（必需）标识自定义菜单要添加到的标准菜单。`menu_name` 的值为以下其中一项：

- 文件
- 编辑
- insert*
- view*
- tools*
- window*
- generate

- help*
- file.project
- file.outputs
- file.models
- edit.stream
- edit.node
- edit.outputs
- edit.models
- edit.project
- tools.repository
- tools.options
- tools.streamProperties

* 仅当添加到主要的 IBM® SPSS® Modeler 窗口时有效

showLabel 指定用户界面上是否显示项目的显示标签。

showIcon 指定用户界面上是否显示项目的相关图标。

separatorBefore 指定是否应将分隔符（例如，菜单项的水平条或工具栏按钮的空格）添加到菜单中新项目的前面。

separatorAfter 指定是否应将分隔符添加到菜单中新项目的后面。

offset 是定义新项目位置的非负整数，例如将 0 偏移量添加它可作为第一个项目（将默认偏移量添加它作为最后一个项目）。

mnemonic 是与 Alt 键一起使用以激活此控件的字母符号（例如，如果您给其值为 S，则用户可通过 Alt-S 激活此控件）。

mnemonicKey 标识助记符以便进行本地化。如果不使用 **mnemonic** 和 **mnemonicKey**，则此控件无可用的助记符。 [有关详细信息，请参阅第 126 页码访问键和键盘快捷键。](#)

菜单项

可以定义自定义菜单项以添加到其中一个标准或自定义菜单中。

格式

```
<MenuItem action="identifier" systemMenu="menu_name" customMenu="menu_name"
  showLabel="true_false" showIcon="true_false" separatorBefore="true_false"
  separatorAfter="true_false" offset="整数" />
```

其中：

action（必需）是在“通用对象”部分定义的操作的标识符，可以指定该菜单项要执行的操作。

systemMenu 指定将出现在由 `menu_name` 标识的标准菜单上的项，它可以是以下任意一项：

- 文件
- 编辑
- insert*
- view*
- tools*
- window*
- generate
- help*
- file.project
- file.outputs
- file.models
- edit.stream
- edit.node
- edit.outputs
- edit.models
- edit.project
- tools.repository
- tools.options
- tools.streamProperties

* 仅当添加到主要的 IBM® SPSS® Modeler 窗口时有效

customMenu 是 **Menu** 元素中的标识符，可以指定在该自定义菜单上出现的菜单项。

showLabel 指定用户界面上是否显示项目的显示标签。

showIcon 指定用户界面上是否显示项目的相关图标。

separatorBefore 指定是否应将分隔符（例如，菜单项的水平条或工具栏按钮的空格）添加到菜单中新项目的前面。

separatorAfter 指定是否应将分隔符添加到菜单中新项目的后面。

offset 是定义新项目位置的非负整数，例如将 0 偏移量添加它可作为第一个项目（将默认偏移量添加它作为最后一个项目）。

示例

```
<MenuItem action="generateSelect" systemMenu="generate" showIcon="true"/>
```

工具栏项

您可以为对话框或输出窗口定义自定义工具栏项。

格式

```
<ToolBarItem action="action" showLabel="true_false" showIcon="true_false"
  separatorBefore="true_false" separatorAfter="true_false" offset="整数" />
```

其中：

action（必需）是在“通用对象”部分定义的操作的标识符，可以指定该工具栏项要执行的操作。

showLabel 指定用户界面上是否显示项目的显示标签。

showIcon 指定用户界面上是否显示项目的相关图标。

separatorBefore 指定是否应将分隔符（例如，菜单项的水平条或工具栏按钮的空格）添加到菜单中新项目的前面。

separatorAfter 指定是否应将分隔符添加到菜单中新项目的后面。

offset 是定义新项目位置的非负整数，例如将 0 偏移量添加它可作为第一个项目（将默认偏移量添加它作为最后一个项目）。

示例

```
<ToolBarItem action="generateDerive" showLabel="true"/>
```

示例：添加至主窗口

这是将一个新项目添加至主窗口中的“工具”菜单的规范文件示例。本示例不定义任何标准对象（例如，节点、模型输出窗口或文档输出窗口），但具有顶层 **Extension**（表示“更改主窗口”）中的 **UserInterface** 元素。所有其他 **UserInterface** 部分都将在其中一个标准对象定义中出现，并且会影响与这些对象相关的对话框。

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0">
  <ExtensionDetails providerTag="example" id="main_window" label="Main Window" version="1.0"
    provider="IBM Corp." copyright="(c) 2005-2006 IBM Corp."
    description="An example extension that plugs into the main window"/>
  <Resources/>
  <CommonObjects>
    <Actions>
      <Action id="customTool1" label="Custom Tool..." labelKey="customTool.LABEL"
```

```

        imagePath="images/generate.gif" description="Invokes the custom tool"
        descriptionKey="customTool.TOOLTIP"/>
    <Action id="customTool2" label="Custom Tool..." labelKey="customTool.LABEL"
        imagePath="images/generate.gif" description="Invokes the custom tool"
        descriptionKey="customTool.TOOLTIP"/>
    </Actions>
</CommonObjects>
<UserInterface actionHandler="com.spss.cleftest.MainWindowActionHandler">
    <Controls>
        <Menu systemMenu="tools" id="toolsExtension" separatorBefore="true"
            label="Extension Items" offset="3"/>
        <MenuItem action="customTool2" customMenu="toolsExtension" showIcon="true"/>
        <MenuItem action="customTool1" systemMenu="file.models" showIcon="true"/>
        <ToolBarItem action="customTool1" offset="0"/>
    </Controls>
</UserInterface>
</Extension>

```

这样做的结果是将名为扩展项的新子菜单添加到“工具”菜单中。该新子菜单有一个名为自定义工具的单个项：

图片 6-4
扩展标准菜单



您可以通过将 XML 代码保存在一个名为 extension.xml 的文件中来尝试本示例，然后将扩展名添加到 CLEF。有关详细信息，请参阅第 219 页码第 10 章中的[测试 CLEF 扩展](#)。

Tabs

Tabs 部分定义了以下位置可能出现的选项卡：

- 当用户打开工作区中的节点时显示的对话框
- 模型输出窗口
- 文档输出窗口

每个 Tabs 部分可以包含多个 Tab 元素，每个元素均可声明一个要显示的自定义选项卡：

```

<Tabs defaultTab="整数">
    <Tab .../>
    <Tab .../>
    ...
</Tabs>

```


其中 `defaultTab` 是一个非负整数，可指定在流中首次打开节点对话框或窗口时要显示的选项卡。当该流处于活动状态时，如果用户在关闭或重新打开对话框或窗口时选择了其他选项卡，则显示最近一次查看的选项卡而非默认选项卡。选项卡从 0 开始计数。

请注意，其他选项卡可能自动包括在对话框或窗口上，例如，所有对话框和输出窗口中均包括注解选项卡，而所有数据源节点对话框中均包括过滤器和类型选项卡。

`Tab` 元素必须声明选项卡标签（选项卡自身上显示的文本），而且还应包含标签键，以便在翻译选项卡标签时用于进行查找。

在每个 `Tab` 元素中有一个面板规范，可以定义选项卡主内容区域的布局方式。面板规范可为以下三种类型之一：文本浏览器、扩展对象或属性。[有关详细信息，请参阅第 127 页码面板规范。](#)

单个 `Tab` 元素的格式为：

```
<Tab id="identifier" label="display_label" labelKey="label_key" helpLink="help_ID"
      mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" >
  <TextBrowserPanel .../>
  <ExtensionObjectPanel .../>
  <PropertiesPanel .../>
  <ModelViewerPanel .../>
</Tab>
```

其中：

`id` 是用于在 Java 代码中引用选项卡的唯一标识符。

`label`（必需）是选项卡在用户界面上的显示名称。

`labelKey` 标识标签以便进行本地化。

`helpLink` 是用户调用帮助系统时显示的帮助主题的标识符（如果有的话）。标识符的格式取决于帮助系统的类型（请参阅[定义帮助系统的位置和类型第 182 页码](#)）：

HTML 帮助：帮助主题的 URL

JavaHelp：主题 ID

`mnemonic` 是与 Alt 键一起使用以激活此控件的字母符号（例如，如果您给其值为 S，则用户可通过 Alt-S 激活此控件）。

`mnemonicKey` 标识助记符以便进行本地化。如果不使用 `mnemonic` 和 `mnemonicKey`，则此控件无可用的助记符。[有关详细信息，请参阅第 126 页码访问键和键盘快捷键。](#)

示例

有关 `Tab` 元素的示例，请参阅不同类型面板规范上的各个部分，其中包括：[文本浏览器面板第 127 页码](#)、[扩展对象面板第 129 页码](#)、[属性面板第 131 页码](#)以及[模型查看器面板第 133 页码](#)。

访问键和键盘快捷键

作为鼠标访问用户界面上功能的替代方式，您可以指定各个按键组合以使用户能通过键盘访问各项功能。

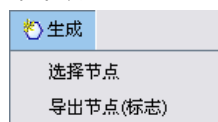
访问键

访问键可与 Alt 键一起使用。对于菜单项、对话框选项卡和各种其他对话框控件，您可以通过使用以下元素的 `mnemonic` 属性指定访问键。

功能	元素	请参阅...
屏幕操作（例如对于菜单项）	<code>action</code>	Actions 第 44 页码
菜单	菜单	菜单 第 120 页码
对话框选项卡	<code>tab</code>	Tabs 第 124 页码
控制器	各种	控制器 第 141 页码

例如，为以下菜单上的项目指定访问键：

图片 6-5
菜单项

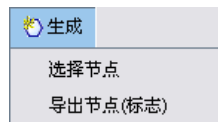


您可以使用以下代码：

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP"mnemonic="S" />
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="生成派生节点"
    descriptionKey="generate.deriveNode.TOOLTIP"mnemonic="D" />
  ...
</Actions>
```

这样便为菜单项上的字符加上下划线：

图片 6-6
使用访问键访问菜单项



用户现在可以通过 Alt-S 和 Alt-D 访问菜单项，也可用鼠标单击进行访问。

快捷键

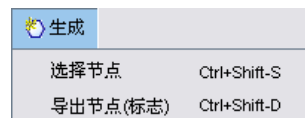
快捷键是使最终用户能在用户界面中导航的按键组合。IBM® SPSS® Modeler 提供多个标准的键盘快捷键。在 CLEF 中，您可以只对您添加的自定义菜单项添加快捷键。

例如，要指定访问键示例中所示菜单的菜单项的快捷键，您可以使用以下代码：

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP" mnemonic="S" shortcut="CTRL+SHIFT+S" />
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="生成派生节点"
    descriptionKey="generate.deriveNode.TOOLTIP" mnemonic="D" shortcut="CTRL+SHIFT+D" />
  ...
</Actions>
```

快捷键组合现在已添加到菜单项：

图片 6-7
使用快捷键访问菜单项



用户现在可以通过键盘快捷键访问菜单项，也可用鼠标单击进行访问。您可以按照该示例来指定快捷键和访问键。

请注意，不要使用已在相同对话框上指定的快捷键，或任一标准 SPSS Modeler 快捷键。有关详细信息，请参阅附录 A 中的键盘辅助功能中的 IBM SPSS Modeler 15 用户指南。

面板规范

每个 Tab 元素可以包含单个面板的规范，这些规范可以是以下类型之一：

表 6-2
面板规范类型

面板	显示...	请参阅...
文本浏览器面板	指定容器的文本内容。	文本浏览器面板第 127 页码
扩展对象面板	由指定的 Java 类定义的内容。	扩展对象面板第 129 页码
属性面板	属性控件（例如，按钮、复选框和输入字段）。	属性面板第 131 页码
模型查看器面板	来自指定容器的 PMML 格式的输出。	模型查看器面板第 133 页码

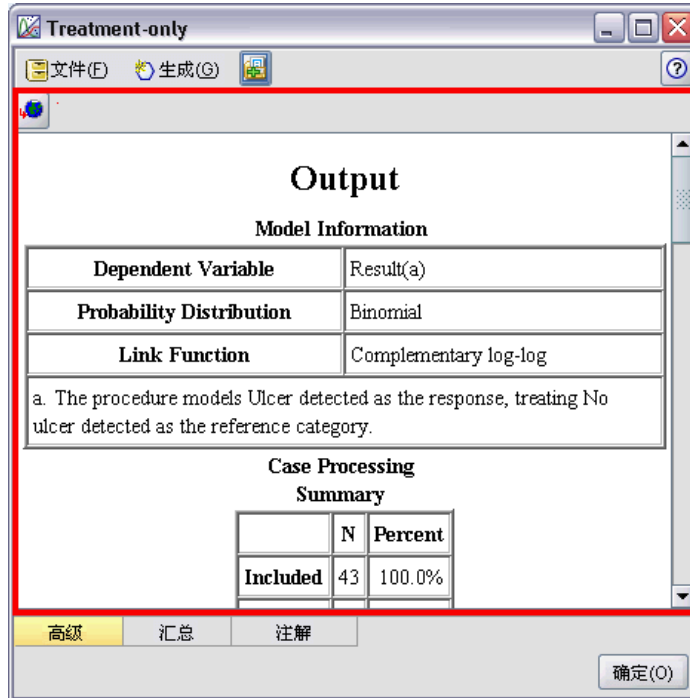
文本浏览器面板

文本浏览器面板显示扩展中指定容器的文本内容。受支持的格式（UTF-8 编码）有纯文本、HTML 和富文本格式（RTF）。

以下示例显示包含 HTML 格式的文本浏览器面板的模型输出窗口：

图片 6-8

突出显示文本浏览器面板的模型输出窗口



格式

```
<TextBrowserPanel container="名称" textFormat="text_format" rows="整数"
  columns="整数" wrapLines="true_false" >
  -- advanced custom layout options --
</TextBrowserPanel>
```

其中：

`container`（必需）是可从中获取面板内容的容器的名称。

`textFormat`（必需）指定面板中所显示文本的格式，具体格式有：

- 纯文本
- html
- rtf

`rows` 和 `columns` 指定当包含面板的窗口打开时显示的文本行数和列数。

`wrapLines` 指定对长文本行是使用换行（`true`），还是需要水平滚动以阅读长文本行（`false`）。默认值为 `false`。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

以下示例演示了“选项卡”部分，该部分定义之前显示的文本浏览器面板：

```
<Tab label="Advanced" labelKey="advancedTab.LABEL" helpLink="genlin_output_advancedtab.htm">
  <TextBrowserPanel container="advanced_output" textFormat="html" />
</Tab>
```

将模型输出发送至容器，该容器在与选项卡规范相同的 `ModelOutput` 部分中定义。

```
<ModelOutput id="generalizedlinear" label="Generalized Linear">
  <Containers>
    ...
    <Container name="advanced_output" />
  </Containers>
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Advanced" ...>
        <TextBrowserPanel container="advanced_output" ... />
      </Tab>
    </Tabs>
  </UserInterface>
</ModelOutput>
```

在“执行”部分的 `CreateDocument` 元素中，为相关构建节点引用该容器：

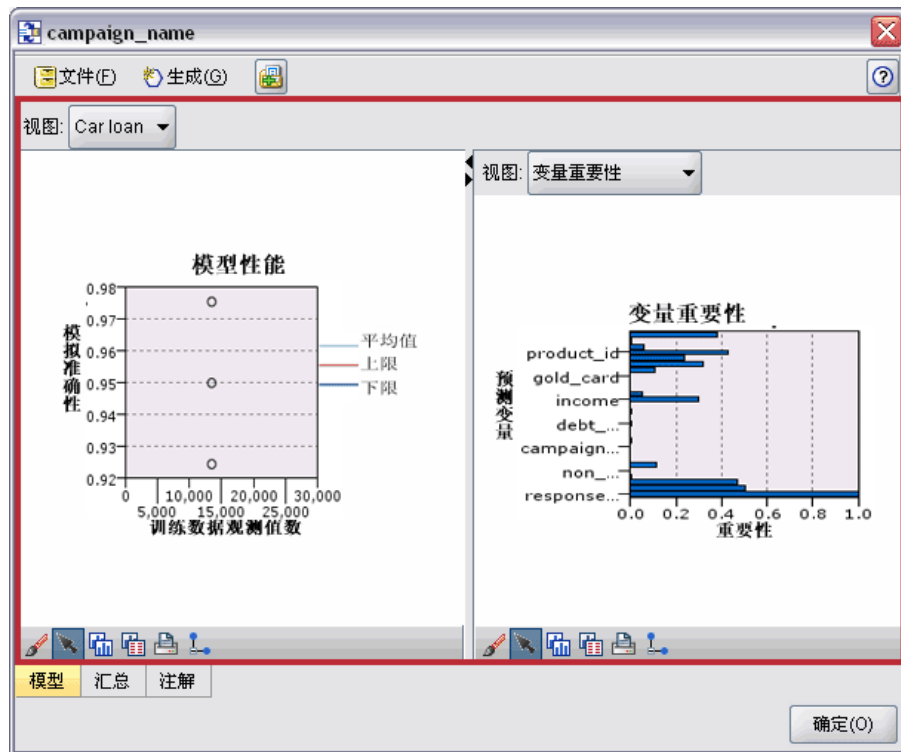
```
<Execution>
  ...
  <Constructors>
    <CreateModelOutput type="generalizedlinear">
      <CreateModel type="generalizedlinear_model" target="model" sourceFile="pmml"/>
      <CreateDocument type="html_output" target="advanced_output" sourceFile="htmloutput"/>
    </CreateModelOutput>
  </Constructors>
</Execution>
```

扩展对象面板

扩展对象面板和文本浏览器面板的工作方式相似，但不显示容器的文本内容，而是创建特定 Java 类的实例，该 Java 类可实现由 CLEF Java API 定义的 `ExtensionObjectPanel` 界面。

以下示例显示包含扩展对象面板的模型应用节点对话框：

图片 6-9
突出显示扩展对象面板的模型输出窗口



格式

```
<ExtensionObjectPanel id="identifier" panelClass="Java_class" >
  -- advanced custom layout options --
</ExtensionObjectPanel>
```

其中：

id 是用于在 Java 代码中引用面板的唯一标识符。

panelClass（必需）是面板所实现的 Java 类的名称。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

以下示例演示了“选项卡”部分，该部分定义之前显示的扩展对象面板：

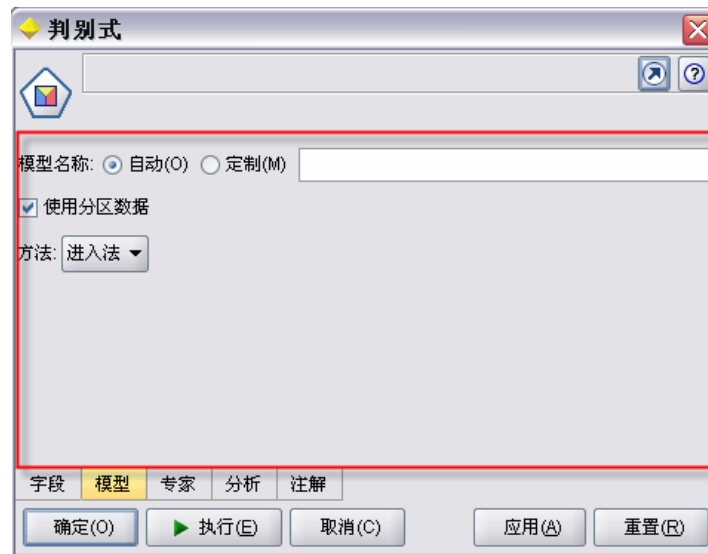
```
<Tab label="Model" labelKey="Model.LABEL" helpLink="selflearnnode_output.htm">
  <ExtensionObjectPanel id="SelfLearningPanel"
    panelClass="com.spss.clef.selflearning.SelfLearningPanel"/>
</Tab>
```

属性面板

属性面板允许选项卡或属性子面板（请参阅[属性子面板第 138 页码](#)）显示**属性控件**，这些控件是一些屏幕组件（如按钮、复选框和输入字段），可用于修改屏幕显示对象的属性。当用户单击**确定**或**应用**后，属性面板将自动应用使用这些控件所作的更改。当用户单击**取消**或**重置**后，面板将放弃自上次应用操作后进行的所有更改。

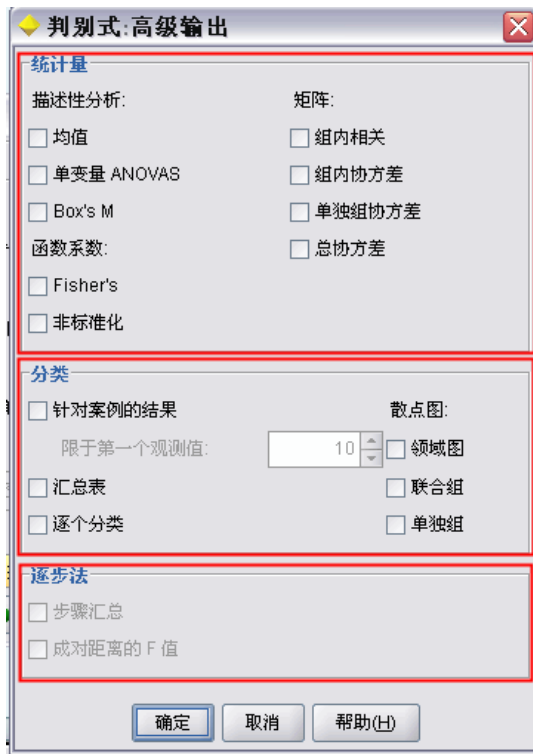
以下示例显示了包含属性面板的节点对话框：

图片 6-10
突出显示属性面板的节点对话框



下一示例演示了包含三个属性面板的属性子面板：

图片 6-11
突出显示属性面板的属性子面板



格式

```
<PropertiesPanel id="identifier" label="display_label" labelKey="label_key">
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesPanel>
```

其中:

`id` 是用于在 Java 代码中引用面板的唯一标识符。

`label` 是属性控件组（例如，上一示例中的统计量、分类和逐步）的显示标题。

`labelKey` 标识标签以便进行本地化。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

有关各个属性控件规范的信息，请参阅[属性控件规范第 134 页码](#)

示例

```
<Tab label="Model" labelKey="Model.LABEL" helpLink="discriminant_node_model.htm">
  <PropertiesPanel>
```



```

<SystemControls controlId="ModelGeneration" />
<ComboBoxControl property="method">
  <Layout fill="none" />
</ComboBoxControl>
</PropertiesPanel>
</Tab>

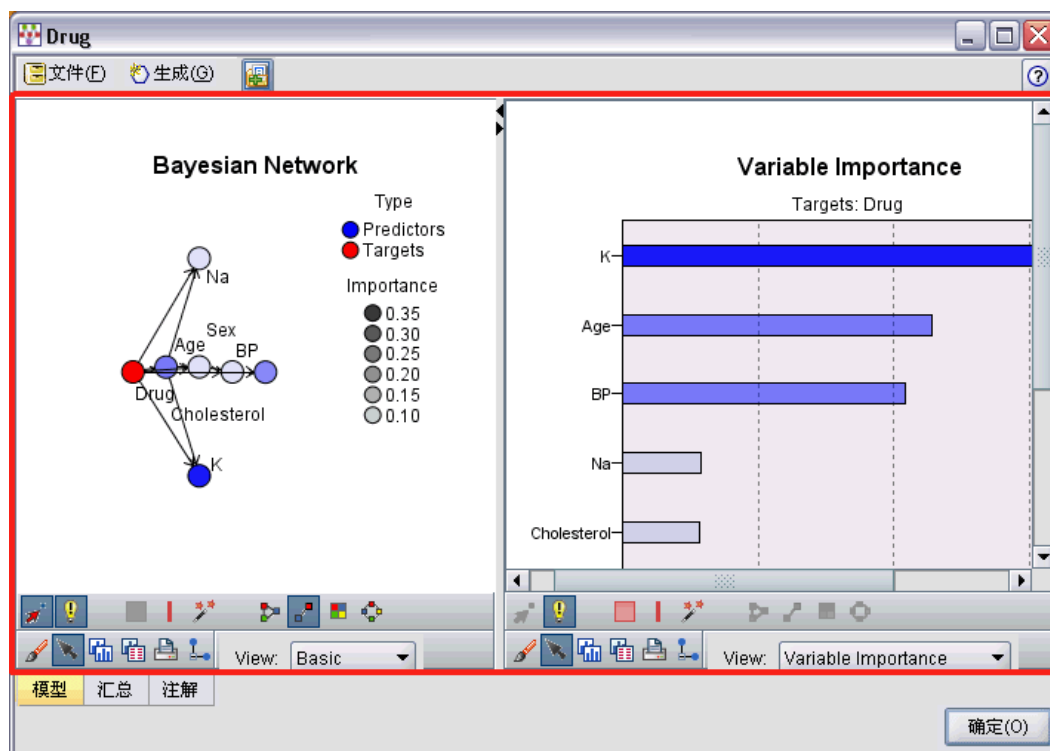
```

模型查看器面板

模型查看器面板可显示扩展中指定容器的任意 PMML 格式的输出。

以下示例显示包含模型查看器面板的模型填充器节点窗口：

图片 6-12
突出显示模型查看器面板的输出窗口



格式

```

<ModelViewerPanel container="container_name">
  -- advanced custom layout options --
</ModelViewerPanel>

```

其中 `container`（必需）是向其中分配模型输出的容器的名称。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。[有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

本示例演示模型查看器面板在模型填充器规范中的用法。该模型输出之前已被分配到名为 `model` 的容器中。此处模型填充器规范将选取此容器，并将其与模型查看器面板相关联：

```
<Node id="applyBN" type="modelApplier">
  <ModelProvider container="模型" isPMML="true" />
  ...
  <Containers>
    <Container name="模型" />
  </Containers>
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Model" labelKey="modelTab.LABEL" helpLink="BN_output_modeltab.htm">
        <ModelViewerPanel container="模型"/>
      </Tab>
    ...
  </Tabs>
</UserInterface>
...
</Node>
```

属性控件规范

属性控件是指一些像按钮、复选框和输入字段这样的屏幕组件，可用于修改屏幕显示对象的属性。属性控件规范的格式由属性控件类型决定，可以是以下类型之一：

- UI 组件
- 属性面板
- 控制器

UI 组件控件可以是操作按钮、显示屏上的静态文本和系统控件（一组对所有对话框都通用的处理属性的控件）。

属性面板控件是指属性面板规范中的各个面板。

控制器可形成最大的属性控件组，并包含许多项，如复选框、组合框和微调控件等。

UI 组件控件

UI 组件控件为：

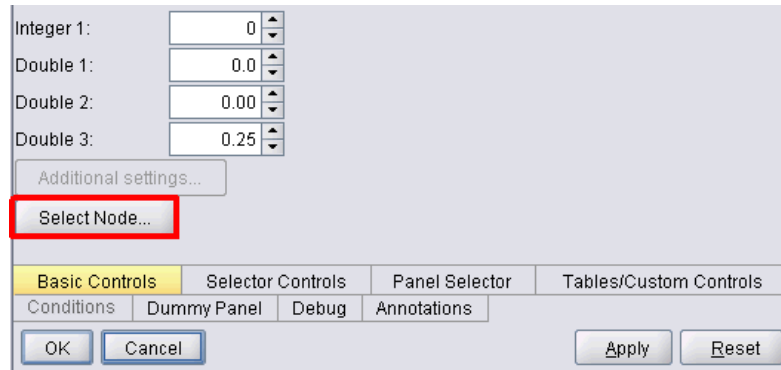
表 6-3
UI 组件控件

控件	说明
ActionButton	屏幕按钮，单击时可执行预定义的操作。
StaticText	显示在屏幕上的非变量文本字符串。
SystemControls	处理所有模型通用属性的标准控件组。

操作按钮

定义可执行“通用对象”部分指定操作的对话框按钮或工具栏按钮。用户单击此按钮后，将执行该操作（例如，显示新屏幕）。

图片 6-13
突出显示操作按钮的对话框



格式

```
<ActionButton action="action" showLabel="true_false" showIcon="true_false" >
  -- advanced custom layout options --
</ActionButton>
```

其中：

action（必需）是要执行的操作的标识符。

showLabel 指定是显示（**true**）按钮标签还是隐藏（**false**）按钮标签（例如，对于工具栏按钮，您可以选择显示图标而不是标签）。默认值为 **true**。

showIcon 指定是显示（**true**）还是隐藏（**false**）与该按钮关联的图标。默认值为 **false**。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

创建之前所显示的操作按钮的代码为：

```
<ActionButton action="generateSelect"/>
```

在“通用对象”部分定义该操作，如下所示（请注意，此处还定义了按钮标签）：

```
<CommonObjects extensionListenerClass="com.spss.cleftest.TestExtensionListener">
  ...
  <Actions>
    <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
      imagePath="images/generate.gif" description="Generates a select node">
```

```

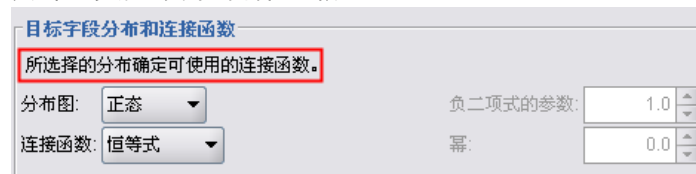
descriptionKey="generate.selectNode.TOOLTIP"/>
...
</Actions>
</CommonObjects>

```

静态文本

通过此元素，您可以在对话框或输出窗口中包含一个非变量文本字符串。以下示例演示了包含静态文本的属性面板：

图片 6-14
突出显示静态文本的属性面板



格式

```

<StaticText text="static_text" textKey="text_key" >
  -- advanced custom layout options --
</StaticText>

```

其中：

`text` 是要使用的文本字符串。

`textKey` 可标识静态文本以供本地化使用。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

以下示例显示之前显示的静态文本所使用的声明：

```

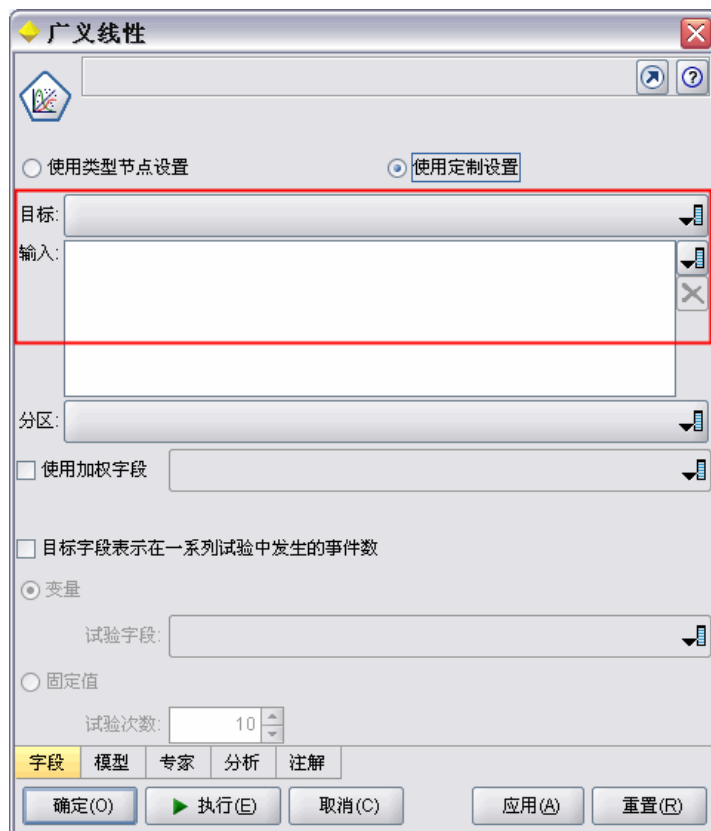
<StaticText text="The distribution that you choose determines which link functions are available."
textKey="Genlin_staticText1"/>

```

系统控件

有些属性所有模型都通用。在模型构建器节点中，系统控件是处理这些属性的标准控件组。

图片 6-15
突出显示系统控件的对话框示例



格式

```
<SystemControls controlsID="identifier" >
  -- advanced custom layout options --
</SystemControls>
```

其中 `controlsID` 是控制组的标识符。此标识符必须与在“模型构建器”声明中的 `ModelingFields` 元素的 `controlsID` 属性中指定的标识符相同（请参阅[模型构建器第 57 页码](#)）。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

本示例显示上一插图中系统控件所使用的声明。

在节点规范的模型构建器部分，以下代码定义了一组系统控件，本示例中包含了模型输入输出字段的字段选取器：

```

<ModelBuilder ...>
  <ModelingFields controlId="modelingFields">
    <InputFields property="inputs" multiple="true" label="Inputs" types="[range set orderedSet flag]"
labelKey="inputFields.LABEL"/>
    <OutputFields property="target" multiple="false" types="[range flag]"
label="Target" labelKey="targetField.LABEL"/>
  </ModelingFields>
  ...
</ModelBuilder>

```

在随后的文件中，该控件组将在它们所在的模型构建器节点对话框的选项卡定义中被引用：

```

<Tab label="Fields" labelKey="Fields.LABEL" helpLink="genlin_node_fieldstab.htm">
  <PropertiesPanel>
    <SystemControls controlId="modelingFields">
      </SystemControls>
    ...
  </PropertiesPanel>
</Tab>

```

属性面板控件

下表列出了属性面板控件：

表 6-4
属性面板控件

控件	说明
PropertiesSubPanel	在用户单击属性面板上的按钮时显示的独立对话框。
PropertiesPanel	嵌套在属性子面板声明或嵌套在顶层属性面板声明中的属性面板。

属性子面板

定义在用户单击属性面板上的按钮时显示的独立对话框。属性子面板声明将成为选项卡的主属性面板规范中的一部分。

格式

```

<PropertiesSubPanel buttonLabel="display_label" buttonLabelKey="label_key"
  dialogTitle="display_title" dialogTitleKey="title_key" helpLink="help_ID"
  mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" >
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesSubPanel>

```

其中：

`buttonLabel` 是单击即可访问子面板的按钮的标签。

`buttonLabelKey` 标识按钮标签以便进行本地化。

`dialogTitle` 是显示在子面板对话框标题栏上的文本。

`dialogTitleKey` 标识子面板对话框标题以便进行本地化。

`helpLink` 是用户调用帮助系统时显示的帮助主题的标识符（如果有的话）。标识符的格式取决于帮助系统的类型（请参阅[定义帮助系统的位置和类型](#)第 182 页码）：

HTML 帮助：帮助主题的 URL

JavaHelp：主题 ID

`mnemonic` 是与 Alt 键一起使用以激活此控件的字母符号（例如，如果您给其值为 S，则用户可通过 Alt-S 激活此控件）。

`mnemonicKey` 标识助记符以便进行本地化。如果不使用 `mnemonic` 和 `mnemonicKey`，则此控件无可用的助记符。有关详细信息，请参阅第 126 页码[访问键和键盘快捷键](#)。

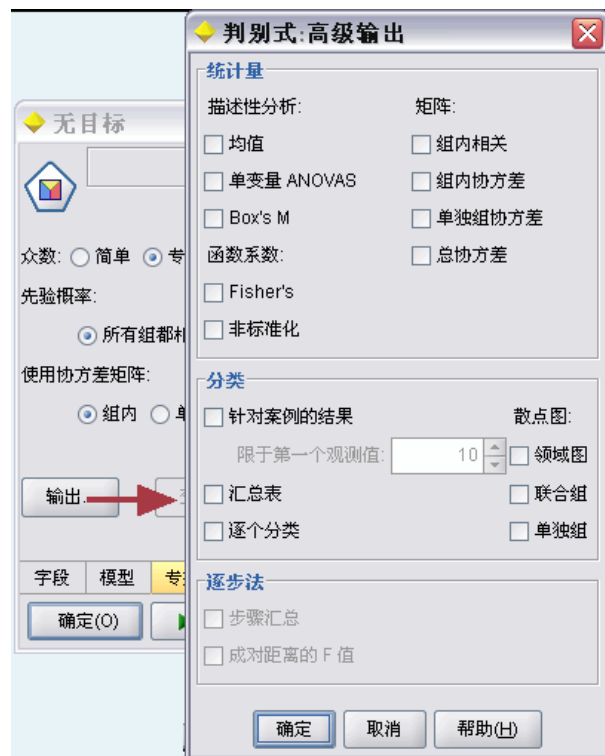
高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。有关详细信息，请参阅第 166 页码[高级自定义布局](#)。

有关各个属性控件规范的信息，请参阅[属性控件规范](#)第 134 页码

示例

以下示例演示了当用户单击选项卡主属性面板上的输出按钮时显示的属性子面板。

图片 6-16
属性子面板



以下代码显示了声明中最重要的部分，该声明可用于获取上一插图中的属性子面板。请注意，在子面板声明中，各个字段组（统计量、分类和逐步）都具有其自身的属性面板规范：

```
<PropertiesSubPanel buttonLabel="Output..." buttonLabelKey="OutputSubPanel.LABEL"
  dialogTitle="Discriminant:Advanced Output" dialogTitleKey="AdvancedOutputSubDialog.LABEL"
  helpLink="discriminant_node_outputdlg.htm">
  ...
  <PropertiesPanel>
    <PropertiesPanel label="Statistics" ...>
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Classification" ...>
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Stepwise" ...>
      ...
    </PropertiesPanel>
  </PropertiesPanel>
</PropertiesSubPanel>
```

属性面板（嵌套）

可以在属性子面板声明中嵌套属性面板规范，以定义从子面板中显示的对话框内容。[有关详细信息，请参阅第 138 页码属性子面板。](#)

也可以在顶层属性面板声明中嵌套属性面板规范。在以下情况下您可能想要执行此操作，根据是否已选择选项卡上的某个特定按钮来启用或禁用整个选项卡（包括多个属性面板）的内容。在此情况下，选项卡规范类似于以下格式：

```
<Tab ....>
  <PropertiesPanel>
    --- button specification ---
    <PropertiesPanel>
      <Enabled>
        --- condition involving button value ---
      </Enabled>
      ...
    </PropertiesPanel>
    <PropertiesPanel>
      <Enabled>
        --- condition involving button value ---
      </Enabled>
      ...
    </PropertiesPanel>
  </PropertiesPanel>
</Tab>
```

嵌套属性面板规范的格式与顶层元素的格式相同。[有关详细信息，请参阅第 131 页码属性面板。](#)

控制器

控制器组成最大的属性控件组：

表 6-5
控制器

控件	说明
CheckBoxControl	复选框。
CheckBoxGroupControl	复选框组，每个 enum 值表示一个复选框。
ClientDirectoryChooserControl	单行文本字段和关联按钮，可允许用户选择客户端上的目录。
ClientFileChooserControl	单行文本字段和关联按钮，可允许用户选择客户端上的文件。
ComboBoxControl	包含 enum 值的下拉组合框。
DBConnectionChooserControl	允许用户选择数据源并连接到数据库。
DBTableChooserControl	允许用户在成功连接到数据库后选择数据库表。
MultiFieldChooserControl	(仅用于节点) 允许用户从中选择一个或多个字段的字段名称列表。
MultiItemChooserControl	允许用户从值列表中选择一项或多项。
PasswordBoxControl	隐藏了输入字符的单行文本字段。
PropertyControl	用户可定义的属性控件。
RadioButtonGroupControl	单选按钮组，一次只能选择一个按钮。对于 enum 属性，每个 enum 值有一个单选按钮；对于布尔属性，则显示两个单选按钮。
ServerDirectoryChooserControl	单行文本字段和关联按钮，可允许用户选择服务器上的目录。
ServerFileChooserControl	单行文本字段和关联按钮，可允许用户选择服务器上的文件。
SingleFieldChooserControl	(仅用于节点) 允许用户从中选择单个字段的字段名称列表。
SingleItemChooserControl	允许用户从值列表中选择单独一项。
SpinnerControl	微调控件 (使用向上和向下箭头更改值的数字字段)。
TableControl	向对话框或窗口中添加表。
TextAreaControl	多行文本字段。
TextBoxControl	单行文本字段。

控制器属性

控制器规范可以包含以下属性：

```
property="value" showLabel="true_false" label="display_label" labelKey="label_key"
labelWidth="label_width" labelAbove="true_false" description="描述"
descriptionKey="description_key" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"
```

其中：

property (必需) 是属性控件的唯一标识符。

`showLabel` 指定是显示 (`true`) 还是隐藏 (`false`) 属性控件的显示标签。默认值为 `true`。

`label` 是属性控件在用户界面上的显示名称。此值还可视为属性控件的简短辅助说明。[有关详细信息，请参阅第 191 页码第 8 章中的辅助功能。](#)

`labelKey` 标识标签以便进行本地化。

`labelWidth` 是标签横跨显示网格列的数目。默认值为 1。

`labelAbove` 指示控件标签是出现在控件上方 (`true`) 还是旁边 (`false`)。默认值为 `false`。

`description` 是光标悬停在控件上时所显示的工具提示文本。此值还可视为属性控件的详细辅助说明。[有关详细信息，请参阅第 191 页码第 8 章中的辅助功能。](#)

`descriptionKey` 标识说明以便进行本地化。

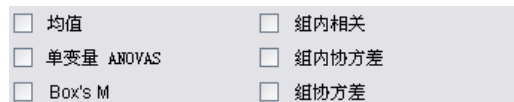
`mnemonic` 是与 Alt 键一起使用以激活此控件的字母符号（例如，如果您给其值为 S，则用户可通过 Alt-S 激活此控件）。

`mnemonicKey` 标识助记符以便进行本地化。如果不使用 `mnemonic` 和 `mnemonicKey`，则此控件无可用的助记符。[有关详细信息，请参阅第 126 页码访问键和键盘快捷键。](#)

复选框控件

定义复选框。

图片 6-17
复选框



格式

```
<CheckBoxControl controller_attributes invert="true_false" >
  -- advanced custom layout options --
</CheckBoxControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`invert` 很少使用，但如果设置为 `true`，则将反转复选框选中和取消选中的效果。默认值为 `false`。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。[有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

以下示例显示用于排列前面所显示的复选框的代码（复选框标签已在规范文件的其他位置定义）。有关 `Layout` 元素的信息，请参阅[高级自定义布局 第 166 页码](#)。

```
<CheckBoxControl property="means">
  <Layout rowIncrement="0" gridWidth="1" />
</CheckBoxControl>
<CheckBoxControl property="within_groups_correlation">
  <Layout gridColumn="2" />
</CheckBoxControl>
<CheckBoxControl property="univariate_anovas">
  <Layout gridWidth="1" rowIncrement="0" />
</CheckBoxControl>
<CheckBoxControl property="within_group_covariance">
  <Layout gridColumn="2" />
</CheckBoxControl>
<CheckBoxControl property="box_m">
  <Layout gridWidth="1" rowIncrement="0" />
</CheckBoxControl>
<CheckBoxControl property="separate_groups_covariance">
  <Layout gridColumn="2" />
</CheckBoxControl>
```

复选框组控件

定义分为一组并作为单个单元处理的复选框集。此控件只能与定义组成员的枚举列表属性一起使用。

图片 6-18
复选框组



格式

```
<CheckBoxGroupControl controller_attributes rows="整数" layoutByRow="true_false"
  useSubPanel="true_false" >
  -- advanced custom layout options --
</CheckBoxGroupControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`rows` 是一个正整数，可指定复选框组在屏幕上所占用的行数。默认值为 1。

`layoutByRow` 指定复选框是先沿着行 (`true`) 还是先顺着列 (`false`) 排列。默认值为 `true`。有关单选按钮组的 `layoutByRow` 的类似用法，请参阅[更改控件顺序第 166 页码](#)。

`useSubPanel` 指定是 (`true`) 否 (`false`) 将复选框作为子面板进行放置。默认值为 `true`。

复选框组通常被作为子面板进行放置，其中包含组中的所有复选框。但是，如果复选框组与相邻的文本字段相关联，则可能会导致出现对齐问题。将 `useSubPanel` 设置为 `false` 可以解决此问题。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

创建之前所显示的复选框组的代码为：

```
<CheckBoxGroupControl property="enum3" label="Enum 3" labelKey="enum3.LABEL"/>
```

与各个复选框相关的标签和值在相应节点的“属性”部分进行定义：

```
<Property name="enum3" valueType="enum" isList="true" defaultValue="[value1 value3]">
  <Enumeration>
    <Enum value="value1" label="Value 3.1" labelKey="enum3.value1.LABEL"/>
    <Enum value="value2" label="Value 3.2" labelKey="enum3.value2.LABEL"/>
    <Enum value="value3" label="Value 3.3" labelKey="enum3.value3.LABEL"/>
    <Enum value="value4" label="Value 3.4" labelKey="enum3.value4.LABEL"/>
    <Enum value="value5" label="Value 3.5" labelKey="enum3.value5.LABEL"/>
  </Enumeration>
</Property>
```

客户端目录选择器控件

定义允许用户选择客户端上目录的单行文本字段和关联按钮。该目录必须已经存在。用户可以从该目录打开文件或将文件保存到该目录中，具体取决于模式设置。

图片 6-19
客户端目录选择器控件



用户可以直接在文本字段中输入目录路径和名称，也可以单击旁边的按钮从所显示的对话框中选择一个目录。

格式

```
<ClientDirectoryChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ClientDirectoryChooserControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`mode` 确定用户可以从中选择目录的对话框中所显示的按钮，可以是以下值之一：

- `open`（默认）显示打开按钮。
- `save` 显示保存按钮。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

```
<ClientDirectoryChooserControl property="directory2" label="Client Directory"
    labelKey="directory2.LABEL"/>
```

客户端文件选择器控件

定义允许用户选择客户端上文件的单行文本字段和关联按钮。该文件必须已经存在。用户可以打开该文件，也可以保存它，具体取决于模式设置。

图片 6-20
客户端文件选择器控件



用户可以在文本字段中直接输入文件路径和名称，也可以单击旁边的按钮从所显示的对话框中选择一个文件。

格式

```
<ClientFileChooserControl controller_attributes mode="chooser_mode" >
    -- advanced custom layout options --
</ClientFileChooserControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`mode` 确定用户可以从对话框中所显示的按钮，可以是以下值之一：

- `open`（默认）显示打开按钮。
- `save` 显示保存按钮。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

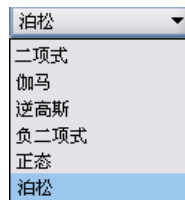
示例

```
<ClientFileChooserControl property="file2" label="Client File" labelKey="file2.LABEL"/>
```

组合框控件

定义组合框下拉列表。

图片 6-21
组合框



格式

```
<ComboBoxControl controller_attributes >
  -- advanced custom layout options --
</ComboBoxControl>
```

其中：

controller_attributes 请参阅[控制器属性第 141 页码](#)。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

下面的示例显示用于设置上一插图中组合框下拉列表布局的代码：

```
<ComboBoxControl property="分布" >
  <Layout rowIncrement="0" gridWidth="1" fill="none"/>
</ComboBoxControl>
```

有关 Layout 元素的信息，请参阅[高级自定义布局 第 166 页码](#)。

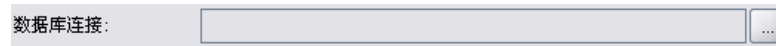
注：实际的列表条目是在相应节点的“属性”部分定义的；以下示例为 distribution 属性声明中的一个枚举列表：

```
<Property name="分布" valueType="enum" label="Distribution" labelKey="distribution.LABEL"
defaultValue="NORMAL">
  <Enumeration>
    <Enum value="BINOMIAL" label="Binomial" labelKey="distribution.BINOMIAL.LABEL"/>
    <Enum value="GAMMA" label="Gamma" labelKey="distribution.GAMMA.LABEL"/>
    <Enum value="IGAUSS" label="Inverse gaussian" labelKey="distribution.IGAUSS.LABEL"/>
    <Enum value="NEGBIN" label="Negative binomial" labelKey="distribution.NEGBIN.LABEL"/>
    <Enum value="NORMAL" label="Normal" labelKey="distribution.NORMAL.LABEL"/>
    <Enum value="POISSON" label="Poisson" labelKey="distribution.POISSON.LABEL"/>
  </Enumeration>
</Property>
```

数据库连接选择器控件

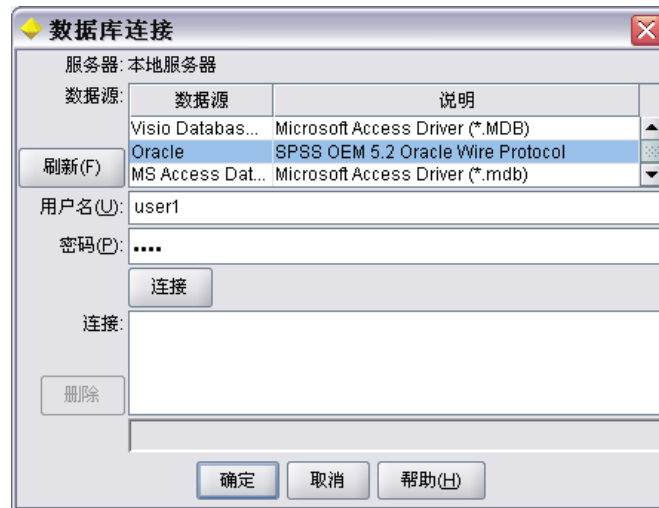
定义使用户可以选择数据源并连接到数据库的控件。

图片 6-22
数据库连接选择器控件



用户不能在该文本字段中输入文本，他们必须单击按钮才能显示标准的 IBM® SPSS® Modeler 数据库连接对话框：

图片 6-23
“数据库连接”对话框



成功连接后，连接详情会显示在数据库连接选择器控件的文本字段中。

格式

```
<DBConnectionChooserControl controller_attributes >
  -- advanced custom layout options --
</DBConnectionChooserControl>
```

其中：

controller_attributes 请参阅[控制器属性第 141 页码](#)。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

以下示例演示控件需要如何定义字符串属性，该字符串属性可以用于连接字符串。

```
<Node ... >
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
  </Properties>
  ...
```

```

<UserInterface>
  ...
  <Tabs>
    <Tab label="Database">
      <PropertiesPanel>
        <DBConnectionChooserControl property="dbconnect" label="Database Connection"/>
        ...
      </PropertiesPanel>
    </Tab>
    ...
  </UserInterface>
</Node>

```

数据库表选择器控件

定义使用户在成功连接数据库后可以选择数据库表的文本字段和关联按钮。

图片 6-24
数据库表选择器控件



用户可以在文本字段中直接输入表名，也可以单击旁边的按钮并从列表中选择所需的表名：

图片 6-25
数据库表列表



格式

```

<DBTableChooserControl connectionProperty="DB_connection_property" controller_attributes >
  -- advanced custom layout options --
</DBTableChooserControl>

```

其中：

`connectionProperty` 是已经定义的数据库连接属性的名称。这是先前为该节点定义的 `DBConnectionChooserControl` 元素的 `property` 属性值。

controller_attributes 请参阅[控制器属性第 141 页码](#)。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

本示例是 `DBConnectionChooserControl` 的一个后续示例，显示如何再包含一个 `DBTableChooserControl` 元素，以在成功建立数据库连接后选择数据库表。

```
<Node ...>
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
    <Property name="dbtable" valueType="string" />
  </Properties>
  ...
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Database">
        <PropertiesPanel>
          <DBConnectionChooserControl property="dbconnect" label="Database connection"/>
          <DBTableChooserControl property="dbtable" connectionProperty="dbconnect"
            label="Database Table" />
          ...
        </PropertiesPanel>
      </Tab>
    </Tabs>
  </UserInterface>
</Node>
```

多字段选择器控件

定义允许用户从列表选择一个或多个字段名的控件。

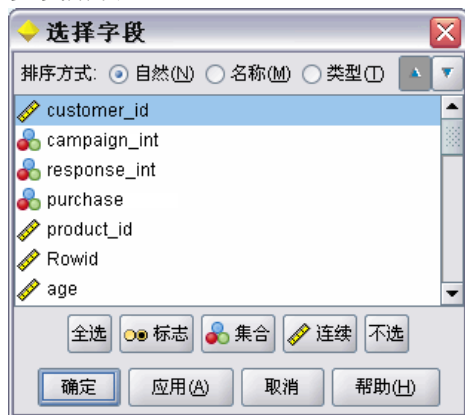
图片 6-26
多字段选择器控件



用户单击此控件时，会显示一个字段列表，用户可以从其中一个或多个字段。

此设置组成了可在此节点中看到的所有字段。如果字段从此节点上游进一步过滤，那么只有通过过滤器的字段才是可见的。该列表可以进一步对之进行限制，其方式是指定仅可选择特定存储和数据类型的字段。

图片 6-27
多字段列表



每一个多字段选择器控件均可指定一个属性特性（该属性特性在文件的其他位置进行声明），并且可定义列表在节点对话框中的显示方式。

格式

```
<MultiFieldChooserControl
controller_attributes storage="storage_types" onlyNumeric="true_false"
    onlySymbolic="true_false" onlyDatetime="true_false" types="data_types"
    onlyRanges="true_false" onlyDiscrete="true_false" >
    -- advanced custom layout options --
</MultiFieldChooserControl>
```

其中：

controller_attributes 请参阅[控制器属性第 141 页码](#)。

此外，还可以通过指定另外两个属性来进一步限制字段列表，其中一个属性必须是下列属性之一：

- **storage** 是列表属性，该属性指定在列表中所允许的字段的存储类型，例如，**storage="[integer real]"** 表示只列出上述存储类型的字段。有关可能存储类型的设置的信息，请参阅[数据和存储类型第 202 页码](#)下的表格。
- **onlyNumeric** 如果设置为 **true**，则指定只列出存储类型为数字的字段。
- **onlySymbolic** 如果设置为 **true**，则指定只列出存储类型为符号（即字符串）的字段。
- **onlyDatetime** 如果设置为 **true**，则指定只列出存储类型为日期和时间的字段。

另一个指定的属性必须来自下表：

- **types** 为列表属性，该属性指定在列表中所允许的字段的日期类型，例如，**types="[range flag]"** 表示只列出上述存储类型的字段。可能的数据类型的设置如下：

范围

标志

```

set
orderedSet
numeric
discrete
typeless

```

- `onlyRanges`，如果设置为 `true`，则指定只列出数据类型为范围的字段。
- `onlyDiscrete` 如果设置为 `true`，则指定只列出数据类型为离散（即标志、设置或无类型）的字段。

因此，例如一个指定 `storage="[integer]"` 和 `types="[flag]"` 的控制确保只有类型为标志的整数字段会显示在列表中。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

注：此控件仅用于 `Node` 元素的定义。要在输出数据模型定义中指定一个多字段选择器，请使用以下格式：

```

<OutputDataModel mode="mode">
...
  <ForEach var="field" inProperty="prop_name">
    <AddField name="{field_name}_NEW" fieldRef="{field_name}" />
  </ForEach>
...
</OutputDataModel>

```

有关详细信息，请参阅第 66 页码第 4 章中的输出数据模型。有关 `ForEach` 元素的信息，请参阅使用 `ForEach` 元素进行迭代第 75 页码。有关 `AddField` 的信息，请参阅添加字段第 72 页码。

示例

下面的示例显示用于指定上一插图中多字段选择器控件的代码。

```

<MultiFieldChooserControl property="输入" >
  <Enabled>
    <Condition control="custom_fields" op="equals" value="true"/>
  </Enabled>
</MultiFieldChooserControl>

```

`Enabled` 部分使该控件仅在选择 `custom_fields` 控件后才能启用。

注：此列表的内容受相应节点属性部分的 `inputs` 属性声明的控制。

```

<Property name="输入" valueType="string" isList="true" label="Inputs" labelKey="inputs.LABEL"/>

```

多项目选择器控件

定义允许用户从值列表中选择一个或多个项目的控件。将属性与包含值列表的目录相关联。有关详细信息，请参阅第 46 页码第 4 章中的Catalogs。

图片 6-28
多项目选择器控件



格式

```
<MultiltemChooserControlcontroller_attributes catalog="catalog_name" >
  -- advanced custom layout options --
</MultiltemChooserControl>
```

其中：

controller_attributes 请参阅控制器属性第 141 页码。

catalog（必需）是要关联的目录的名称。从中获取目录的库在“执行”部分的 Module 元素中指定。有关详细信息，请参阅第 64 页码第 4 章中的模块。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。有关详细信息，请参阅第 166 页码高级自定义布局。

示例

```
<MultiltemChooserControl property="selection2" catalog="cat2" />
```

property 属性（在本例中为 selection2）引用的属性必须具有 isList="true" 特性。有关 MultiltemChooserControl 用法的解释和示例，请参阅Catalogs第 46 页码。

密码框控件

定义可以隐藏输入字符的单行文本字段。

图片 6-29
密码框控件



格式

```
<PasswordBoxControl controller_attributes columns="整数" >
  -- advanced custom layout options --
</PasswordBoxControl>
```

其中：

controller_attributes 请参阅控制器属性第 141 页码。

`columns` 是一个正整数，定义密码框应占用的字符列数。默认值为 20。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

```
<PasswordBoxControl property="encrypted_string1" label="Encrypted string
1" labelKey="encryptedString1.LABEL"/>
```

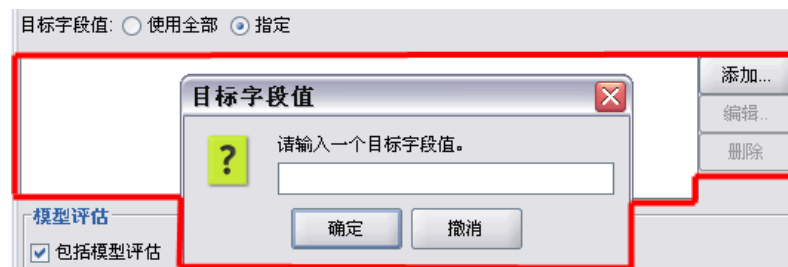
可以采用与相应节点的属性部分中定义为加密字符串的属性相关联这样的方式对文本字段加密：

```
<Property name="encrypted_string1" valueType="encryptedString"/>
```

属性控件

属性控件是一个完全可由用户来定义的控件，通过该控件用户可以输入节点的属性。处理操作由用户编写的 Java 类执行。以下是属性控件的一个示例：

图片 6-30
突出显示属性控件示例的对话框部分



格式

```
<PropertyControl controller_attributes controlIdClass="Java_class" >
  -- advanced custom layout options --
</PropertyControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`controlClass` (必需) 是 .jar 文件内指向实现属性控件的 Java 类的路径。(注：.jar 文件在资源部分的 `JarFile` 元素内声明。 [有关详细信息，请参阅第 39 页码第 4 章中的 Jar 文件。](#))

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

```
<PropertyControl property="target_field_values_specify" labelAbove="true"
  controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" label=""
  labelKey="target_field_values_specify.LABEL">
  <Enabled>
    <Condition control="target_field_values" op="equals" value="specify"/>
  </Enabled>
  <Layout rowIncrement="2" />
</PropertyControl>
```

属性控件与相应节点的属性部分定义的一个属性相关联：

```
<Property name="target_field_values_specify" valueType="string" isList="true" label=""
  labelKey="target_field_values_specify.LABEL"/>
```

单选按钮组控件

定义一组单选按钮，一次仅能选择一个按钮。

图片 6-31

单选按钮组控件

分类输入值的顺序: 升序 降序 使用数据原有顺序

每一个单选按钮组控件均有一个将该组与特定属性相关联的属性特性。此属性是在文件的其他位置定义的，可指定组成该组的按钮。

关联的属性可以是一个枚举列表，也可以是一个布尔属性。对于枚举列表（其中属性特性为 `valueType="enum"`），显示的每一个单选按钮均对应于一个 `enum` 值。对于布尔属性（其中 `valueType="boolean"`），始终显示两个单选按钮。

格式

```
<RadioButtonGroupControl controller_attributes
  rows="整数" layoutByRow="true_false" useSubPanel="true_false"
  falseLabel="button_label" falseLabelKey="label_key" trueLabel="?button_label"
  trueLabelKey="label_key" trueFirst="true_false" >
  -- advanced custom layout options --
</RadioButtonGroupControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`rows` 是一个正整数，可指定单选按钮组显示时要占用的屏幕行数。默认值为 1。

`layoutByRow` 指定单选按钮是先沿着行 (`true`) 还是先顺着列 (`false`) 排列。默认值为 `true`。有关单选按钮组的 `layoutByRow` 用法的示例，请参阅[更改控件顺序第 166 页码](#)。

`useSubPanel` 指定是 (`true`) 否 (`false`) 将单选按钮作为子面板进行放置。默认值为 `true`。

单选按钮通常被作为子面板放置，其中包含组中的所有按钮。但是，如果单选按钮与相邻的文本字段相关联，则可能会导致出现对齐问题。将 `useSubPanel` 设置为 `false` 可以解决此问题。

`falseLabel` 是布尔属性“false”值的标签（请参阅下面的第二个示例）。仅能与布尔属性一起使用，在此情况下它是必需的。

`falseLabelKey` 标识“false”标签以便进行本地化。

`trueLabel` 是布尔属性“true”值的标签（请参见下面的第二个示例）。仅能与布尔属性一起使用，在此情况下它是必需的。

`trueLabelKey` 标识“true”标签以便进行本地化。

`trueFirst`，如果设置为 `true`，会反转布尔属性按钮的显示顺序，因此会先显示对应于“true”值的按钮。默认值为 `false`，意味着会先显示对应于“false”值的按钮。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。[有关详细信息，请参阅第 166 页码高级自定义布局。](#)

示例

第一个示例演示之前显示的单选按钮组所使用的代码。

```
<RadioButtonGroupControl property="value_order" labelWidth="2">
  <Layout gridWidth="4"/>
</RadioButtonGroupControl>
```

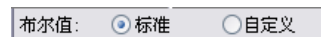
有关 `Layout` 元素的信息，请参阅[高级自定义布局 第 166 页码](#)。

注：按钮数及其标签是在相应节点的属性部分定义的；以下示例为 `value_order` 属性声明中的一个枚举列表。此声明还包括单选按钮组本身的标签：

```
<Property name="value_order" valueType="enum" label="Value order for categorical
  inputs" labelKey="value_order.LABEL">
  <Enumeration>
    <Enum value="Ascending" label="Ascending" labelKey="value_order.Ascending.LABEL"/>
    <Enum value="Descending" label="Descending" labelKey="value_order.Descending.LABEL"/>
    <Enum value="DataOrder" label="Use data order" labelKey="value_order.UseDataOrder.LABEL"/>
  </Enumeration>
</Property>
```

第二个示例演示如何对控制布尔属性的单选按钮组使用 `falseLabel` 和 `trueLabel`，如一个控制启用标准还是自定义设置：

图片 6-32
控制布尔属性的单选按钮组



实现上述目的的代码是：

```
<RadioButtonGroupControl property="boolean5" label="Boolean 5" labelKey="boolean5.LABEL"
  falseLabel="Standard" falseLabelKey="boolean5.false.LABEL" trueLabel="Custom"
  trueLabelKey="boolean5.true.LABEL" />
```

在这种情况下，在 `RadioButtonGroupControl` 元素本身中定义按钮标签和组标签。在节点的属性部分中定义与组相关联的属性：

```
<Property name="boolean5" valueType="boolean" defaultValue="false"/>
```

服务器目录选择器控件

定义允许用户选择服务器上目录的单行文本字段和相关按钮。该目录必须已经存在。用户可以从该目录打开文件或将文件保存到该目录中，具体取决于模式设置。

图片 6-33
服务器目录选择器控件



用户可以直接在文本字段中输入目录路径和名称，也可以单击旁边的按钮从所显示的对话框中选择一个目录。

格式

```
<ServerDirectoryChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ServerDirectoryChooserControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`mode` 确定用户可以从中选择目录的对话框中所显示的按钮，可以是以下值之一：

- `open`（默认）显示打开按钮。
- `save` 显示保存按钮。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

```
<ServerDirectoryChooserControl property="directory1" label="Server Directory"
  labelKey="directory1.LABEL"/>
```

服务器文件选择器控件

定义允许用户选择服务器上文件的单行文本字段和相关按钮。该文件必须已经存在。用户可以打开该文件，也可以保存它，具体取决于模式设置。

图片 6-34
服务器文件选择器控件



用户可以在文本字段中直接输入文件路径和名称，也可以单击旁边的按钮从所显示的对话框中选择一个文件。

格式

```
<ServerFileChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ServerFileChooserControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`mode` 确定用户可以从中选择文件的对话框中所显示的按钮，可以是以下值之一：

- `open`（默认）显示打开按钮。
- `save` 显示保存按钮。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

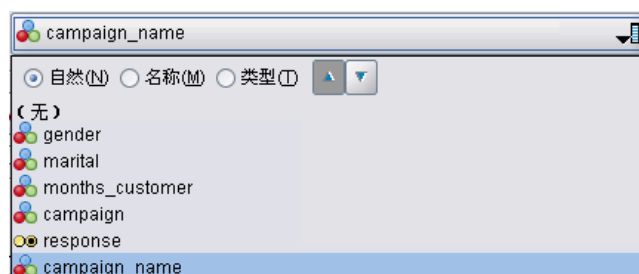
示例

```
<ServerFileChooserControl property="file1" label="Server File" labelKey="file1.LABEL"/>
```

单字段选择器控件

定义允许用户从列表中选择单个字段的控件。

图片 6-35
单个字段选择器控件



当用户单击此控件时，会显示一个字段列表，他们可以从该列表中选择单个字段。

此设置组成了可在此节点中看到的所有字段。如果字段从此节点上游进一步过滤，那么只有通过过滤器的字段才是可见的。该列表可以进一步对之进行限制，其方式是指定仅可选择特定存储和数据类型的字段。

格式

```
<SingleFieldChooserControl
controller_attributes storage="storage_types" onlyNumeric="true_false"
    onlySymbolic="true_false" onlyDatetime="true_false" types="data_types"
    onlyRanges="true_false" onlyDiscrete="true_false" >
    -- advanced custom layout options --
</SingleFieldChooserControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

此外，还可以通过指定另外两个属性来进一步限制字段列表，其中一个属性必须是下列属性之一：

- `storage` 是列表属性，该属性指定在列表中所允许的字段的存储类型，例如，`storage="[integer real]"` 表示只列出上述存储类型的字段。有关可能存储类型的设置的信息，请参阅[数据和存储类型第 202 页码](#)下的表格。
- `onlyNumeric` 如果设置为 `true`，则指定只列出存储类型为数字的字段。
- `onlySymbolic` 如果设置为 `true`，则指定只列出存储类型为符号（即字符串）的字段。
- `onlyDatetime` 如果设置为 `true`，则指定只列出存储类型为日期和时间的字段。

另一个指定的属性必须来自下表：

- `types` 为列表属性，该属性指定在列表中所允许的字段的日期类型，例如，`types="[range flag]"` 表示只列出上述存储类型的字段。可能的数据类型的设置如下：

范围

标志

set

orderedSet

numeric

discrete

typeless

- `onlyRanges`，如果设置为 `true`，则指定只列出数据类型为范围的字段。
- `onlyDiscrete` 如果设置为 `true`，则指定只列出数据类型为离散（即标志、设置或无类型）的字段。

因此，例如一个指定 `storage="[integer]"` 和 `types="[flag]"` 的控制确保只有类型为标志的整数字段会显示在列表中。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

注：此控件仅用于节点定义。要在输出数据模型定义中指定一个多字段选择器，请使用以下格式：

```
<OutputDataModel mode="mode">
...
  <ForEach var="field" from="1" to="{整数}">
    <AddField name="{字符串}_{字段}" fieldRef="{field_ref}" />
  </ForEach>
...
</OutputDataModel>
```

有关详细信息，请参阅第 66 页码第 4 章中的输出数据模型。有关 ForEach 元素的信息，请参阅使用 ForEach 元素进行迭代第 75 页码。有关 AddField 的信息，请参阅添加字段第 72 页码。

示例

下面的示例显示用于指定上一插图中单个字段选择器控件的代码。

```
<SingleFieldChooserControl property="target" storage="string" onlyDiscrete="true"/>
```

注：列表的实际内容是在相应节点的属性部分定义的；在本示例中位于 target 属性的声明中：

```
<Property name="target" valueType="string" label="Target field" labelKey="target.LABEL"/>
```

单项目选择器控件

定义允许用户从值列表中选择单个项目的控件。将属性与包含值列表的目录相关联。有关详细信息，请参阅第 46 页码第 4 章中的 Catalogs。

图片 6-36
单项目选择器控件



格式

```
<SingleItemChooserControlcontroller_attributes catalog="catalog_name" >
  -- advanced custom layout options --
</MultiItemChooserControl
```

其中：

controller_attributes 请参阅控制器属性第 141 页码。

catalog（必需）是要关联的目录的名称。从中获取目录的库在“执行”部分的 Module 元素中指定。有关详细信息，请参阅第 64 页码第 4 章中的模块。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。有关详细信息，请参阅第 166 页码高级自定义布局。

示例

```
<SingleItemChooserControl property="selection1" catalog="cat1" />
```

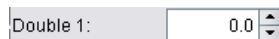
有关该控件用法的解释和示例，请参阅[Catalogs第 46 页码](#)。

微调控件

定义微调框（使用上下箭头更改字段值的数字字段）。

图片 6-37

微调框



格式

```
<SpinnerControlcontroller_attributes columns="整数" stepSize="increment"
  minDecimalDigits="数字" maxDecimalDigits="数字" >
  -- advanced custom layout options --
</SpinnerControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`columns` 是一个正整数，指定控件所跨越的字符列数。默认值为 5。

`stepSize` 是一个十进制数，指定用户单击其中一个箭头时字段值的更改量。默认值为 1.0。

`minDecimalDigits` 是字段值所显示的最小小数位数。默认值为 1。

`maxDecimalDigits` 是字段值所显示的最大小数位数。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

下面的示例显示用于指定上一插图中微调控件的代码：

```
<SpinnerControl property="double1" label="Double 1" labelKey="double1.LABEL"/>
```

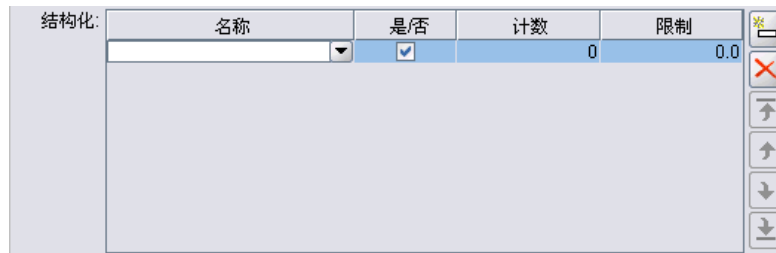
该数字字段内容的精确度和有效范围是在相应节点的属性部分定义的；在本示例中位于 `double1` 属性的声明中：

```
<Property name="double1" valueType="double" min="0" max="100"/>
```

表控件

定义要显示在节点对话框或输出窗口中的表布局项目。

图片 6-38
表控件



格式

```
<TableControl controller_attributes rows="整数" columns="整数" columnWidths="list" >
  -- advanced custom layout options --
</TableControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`rows` 是一个正整数，指定屏幕上显示的表行数。默认值为 8。

`columns` 是一个正整数，指定表所跨越的字符列数。默认值为 20。

`columnwidths` 是指定相应列宽的一列值。因此，如果值为 [30 5 10]，则说明第 1 列的宽度是第 3 列的三倍。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

以下示例是用于指定上一插图中表控件的代码：

```
<TableControl property="structure1" allowReorder="true" label="Structured"
  labelKey="structure1.LABEL" columnWidths="[20 6 10 10]">
  <ColumnControl column="0" editor="fieldValue" fieldControl="field1"/>
</TableControl>
```

表控件的结构在规范文件的“通用对象”部分定义为属性类型：

```
<PropertyType id="shared_structure1" valueType="structure" isList="true">
  <Structure>
    <Attribute name="id" valueType="string" label="Name" labelKey="structure1.id.LABEL"/>
    <Attribute name="yesno" valueType="boolean" label="Yes/No" labelKey="structure1.yesno.LABEL"
      defaultValue="true"/>
    <Attribute name="count" valueType="integer" label="Count" labelKey="structure1.count.LABEL"
      defaultValue="0"/>
    <Attribute name="limit" valueType="double" label="Limit" labelKey="structure1.limit.LABEL"
      defaultValue="0.0"/>
  </Structure>
```

```
</PropertyType>
```

在节点规范中，此属性类型的标识符随后通过属性声明与表控件的标识符相关联：

```
<Property name="structure1" type="shared_structure1"/>
```

如果引用脚本中的节点，则可以使用方括号 [] 为列表设置属性值，或使用大括号 {} 为结构设置属性值。例如，可以为 `structure1` 属性设置一个由两个结构组成的网格：

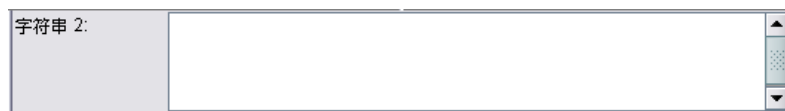
```
set:node_ID.structure1 = [{"hello" true 4 0.21} {"bye" false 5 0.95}]
```

请注意，值的顺序必须与 `Attribute` 的定义顺序一致。

文本区域控件

定义多行文本输入字段。

图片 6-39
文本区域控件



格式

```
<TextAreaControl controller_attributes rows="整数" columns="整数" wrapLines="true_false" >
  -- advanced custom layout options --
</TextAreaControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`rows` 是一个正整数，指定文本区域占用的屏幕行数。默认值为 8。

`columns` 是一个正整数，指定文本区域所跨越的字符列数。默认值为 20。

`wrapLines` 指定对长文本行是使用换行 (`true`)，还是需要水平滚动以阅读长文本行 (`false`)。默认值为 `true`。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

用于创建之前所示示例的代码如下所示：

```
<TextAreaControl property="string2" label="String 2" labelKey="string2.LABEL"/>
```

在本示例中，文本区域的标签定义在文本区域控件声明中，而输入数据类型则定义在相关节点的属性部分，位于 `string2` 属性的声明中：

```
<Property name="string2" valueType="string" />
```

文本框控件

定义单行文本输入字段。

图片 6-40
文本框控件



格式

```
<TextBoxControl controller_attributes columns="整数" >
  -- advanced custom layout options --
</TextBoxControl>
```

其中：

`controller_attributes` 请参阅[控制器属性第 141 页码](#)。

`columns` 是一个正整数，指定文本框所跨越的字符列数。默认值为 20。

高级自定义布局选项对屏幕组件的配置和显示提供了精度控制。 [有关详细信息，请参阅第 166 页码高级自定义布局](#)。

示例

用于创建之前所示文本框的代码如下所示：

```
<TextBoxControl property="string1" label="String 1" labelKey="string1.LABEL"/>
```

文本框的输入数据类型在相应节点的属性部分定义；在本示例中位于 `string1` 属性的声明中：

```
<Property name="string1" valueType="string" />
```

属性控件布局

本节介绍用于对话框和窗口的标准布局方法，以及对标准布局方法进行修改以获得自定义布局的方法。

标准控件布局

可将属性面板视为单元格的二维网格。每一行均可以有不同的高度，每一列均可以有不同的宽度。UI 组件可以分配给多个连续的单元格，但通常情况下每个单元格仅分配一个 UI 组件。

默认情况下，将一个属性控件分配给一行，且每一个控件占两列：其中一列用于标签，另一列用于控件组件。包含标签的列宽度会扩展到最宽标签的宽度。例如，在规范文件中给出以下元素：

```
<TextBoxControl property="string1" label="String 1"/>  
<PasswordBoxControl property="encryptedString1" label="Encrypted string 1"/>  
<TextAreaControl property="string2" label="String 2"/>
```

所得的结果面板如下所示：

图片 6-41
简单的属性面板

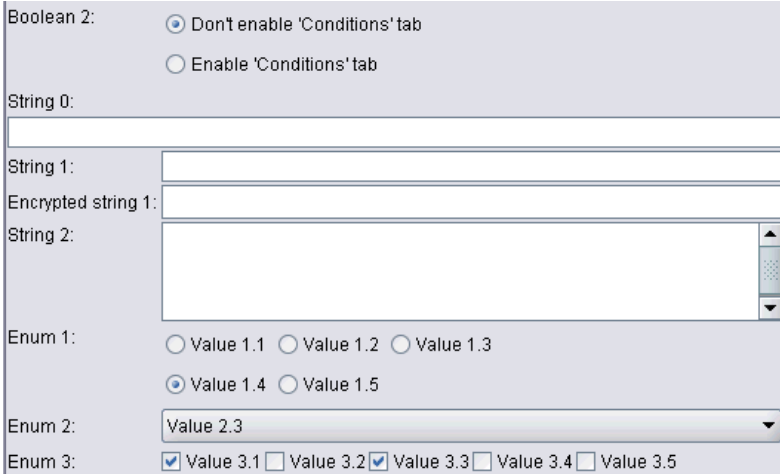


The screenshot shows a property panel with three rows. The first row is labeled '字符串 1:' and has a single-line text input field. The second row is labeled '加密字符串 1:' and has a single-line text input field. The third row is labeled '字符串 2:' and has a multi-line text area with a vertical scrollbar on the right side.

请注意，标签末尾的“:”字符是自动添加的。

包含多个用户界面组件的属性控件创建自己的不可视矩形区域以放置这些组件。**RadioButtonGroupControl** 和 **CheckBoxGroupControl** 元素是这种组件的示例。在以下屏幕视图中，标有 **Boolean2**、**Enum1** 和 **Enum3** 的控件说明了这一点：

图片 6-42
带有各种多组件控件的属性面板

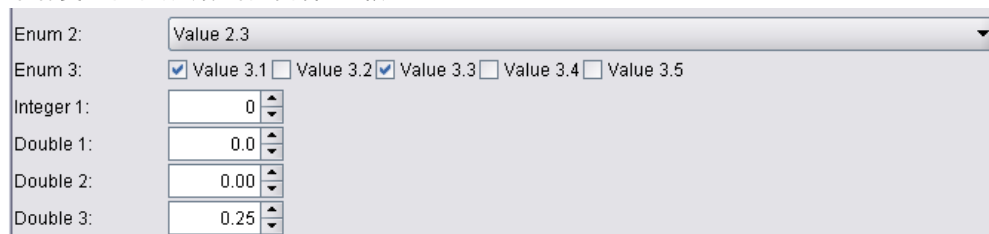


The screenshot shows a complex property panel with several sections. At the top, 'Boolean 2:' has two radio buttons: 'Dont enable 'Conditions' tab' (selected) and 'Enable 'Conditions' tab'. Below that, 'String 0:' has a single-line text input field. 'String 1:' has a single-line text input field. 'Encrypted string 1:' has a single-line text input field. 'String 2:' has a multi-line text area with a vertical scrollbar. 'Enum 1:' has five radio buttons: 'Value 1.1', 'Value 1.2', 'Value 1.3', 'Value 1.4' (selected), and 'Value 1.5'. 'Enum 2:' has a dropdown menu showing 'Value 2.3'. 'Enum 3:' has five checkboxes: 'Value 3.1' (checked), 'Value 3.2', 'Value 3.3' (checked), 'Value 3.4', and 'Value 3.5'.

注意，根据属性控件，放置组件的矩形区域形状可能有所不同。因此，不同控件的布局可能不会总是严格对齐——在前面说明中比较 **Enum 1** 和 **Enum 3**。

一些属性控件包括完全填充组件列的组件，当窗口宽度放大或缩小时水平调整大小。例如，**TextBoxControl**、**PasswordBoxControl** 和 **TextAreaControl** 元素指定的控件，如前面说明中标有 **String 1**、**Encrypted string 1** 和 **String 2** 的控件所示。但是，并非所有组件都可以执行此操作。例如，复选框和微调控件仅占用固定的水平空间量，甚至当窗口宽度放大时也是如此：

图片 6-43
带有复选框和微调控件的属性面板



自定义控件布局

可以采用多种方式修改控件的标准布局，其中有些方法比较简单，而有些方法比较复杂。

简单自定义布局

自定义控件布局有以下三种简单方法：

- 将标签放置到其组件的上方
- 更改控件占用的行数
- 更改控件的放置顺序

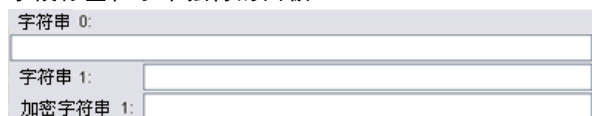
将标签放置到其组件的上方

可以通过将控件的 `labelAbove` 属性设置为 `true` 来将标签放置到其组件上方的单独一行。例如：

```
<TextBoxControl property="string0" label="String 0" labelAbove="true"/>
<TextBoxControl property="string1" label="String 1"/>
<PasswordBoxControl property="encryptedString1" label="Encrypted string 1"/>
```

将标签放置到其组件上方的同时，实际的 UI 组件或组件会被分配到标签显示列。这会生成如下所示的面板，其中标签 `String 0` 将显示在其对应字段的上方：

图片 6-44
字段标签位于单独行的面板



更改行数

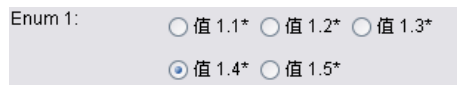
缺省情况下，在单行中放置单选按钮和复选框组，并调整对话框的宽度以容纳它们。如果单选按钮或复选框组有大量选项，可能导致对话框非常宽。您可以通过更改用于放置控件的行数来避免这一情况。通过将控件定义的 `rows` 属性设置为所需的值，可以实现上述配置。例如：

```
<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2"/>
```

这会生成如下所示的面板，其中单选按钮组占用了两行：

图片 6-45

单选按钮组占用两行的面板



更改控件顺序

对于单选按钮组和复选框组，还可以针对每一个 `enum` 值更改控件添加到面板的顺序。

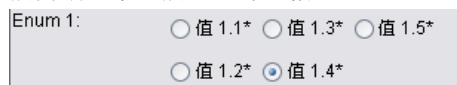
默认情况下，按行添加控件，如上一示例所示，前三个值添加到了第一行，而第四、第五个值添加到了第二行。另外，可以通过将 `layoutByRow` 设置为 `false` 来在指定的行数内按列添加控件。例如：

```
<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2" layoutByRow="false"/>
```

值仍然显示在两行中，但现在第一和第二个值添加到了第一列，第三和第四个值添加到了第二列，第五个值添加到了第三列：

图片 6-46

按列放置单选按钮组的面板



对于以两个单选按钮显示的布尔属性，默认的顺序行为是“False”按钮在前，“True”按钮在后。可以通过将 `trueFirst` 属性设置为 `true` 来颠倒此顺序。

还可以通过将 `useSubPanel` 属性设置为 `false` 来禁止单选按钮组和复选框组使用子面板。但是，这会产生一些不理想的布局行为，除非上述用法与 `Layout` 元素一起使用（请参阅[使用 Layout 元素指定精确的控件位置第 166 页码](#)）。

高级自定义布局

在每个控件声明中，您可以使用各种元素指定复杂的控件布局。可以进行下列操作：

- 使用 `Layout` 元素指定控件在屏幕上的精确位置
- 使用 `Enabled` 元素控制显示特征
- 使用 `Visible` 元素控制屏幕组件的可见性

使用 Layout 元素指定精确的控件位置

通过指定明确的 `Layout` 元素并将其与控件相关联可以实现精确的布局定位：

格式

```
<property_control ... >
  <Layoutattributes
    --- cell specification ---
    ...
```

```
</property_control>
```

其中：

`property_control` 是属性控件之一（请参阅[属性控件规范第 134 页码](#)）。

`attributes` 可以为以下任意属性：

表 6-6
布局属性

属性	值	说明
<code>anchor</code>	north northeast east southeast south southwest west northwest center	定义控件的定位点。
<code>columnWeight</code>	0.0 - 1.0	定义窗口在水平方向发生的大小变化会如何影响控件的宽度。面板中所有 <code>columnWeight</code> 属性的总和不应超过 1.0。
<code>fill</code>	none horizontal vertical both	定义控件是否以及如何填充分配给它的单元格。
<code>gridColumn</code>	integer ≥ 0	定义开始放置控件的第一列。
<code>gridHeight</code>	整数	定义放置控件时控件应占用的行数。值 0（默认值）会跨所有剩余的行分配控件。
<code>gridRow</code>	integer ≥ 0	定义放置控件的第一行。默认情况下，网格行索引会自动增加。
<code>gridWidth</code>	整数	定义放置控件时控件应占用的列数。值 0（默认值）会跨所有剩余的列分配控件。
<code>leftIndent</code>	整数	定义控件相对于其默认位置的缩进像素数。
<code>rowWeight</code>	0.0 - 1.0	定义窗口在垂直方向发生的大小变化会如何影响控件的高度。面板中所有 <code>rowWeight</code> 属性的总和不应超过 1.0。

通过 `cell specification` 可以指定控件在屏幕上的精确位置。格式如下所示：

```
<Cell row="整数" column="整数" width="整数" />
```

其中：

`row`（必需）是一个非负整数，可指定控件的起始行位置。

`column`（必需）是一个非负整数，可指定控件的起始列位置。

`width`（必需）是一个非负整数，可指定控件占用的屏幕网格列数。

因此，假设有一个三列三行的屏幕网格，以下面的格式自定义控件布局：

图片 6-47
使用单元格的控件布局示例

	列 0	列 1	列 2
行 0	控制 1		
行 1	控制 2		
行 2		控制 3	

需要 `Layout` 元素具有以下单元格规范：

```
<Layout ...>
  <Cell row="0" column="0" width="2">
  <Cell row="1" column="0" width="1">
  <Cell row="2" column="0" width="3">
</Layout>
```

下面有一些详细的示例将进一步说明如何使用 `Layout` 元素。

示例：启用文本字段的复选框

本示例演示如何在使用复选框时使文本字段位于相同的显示行中。

使用复选框时要使另一控件位于相同的行中，需要使用一个简单的 `Layout` 元素以正确显示该控件。（注：有关启用和禁用控件的机制，请参阅[使用 Enabled 元素控制显示特征第 173 页码](#)）。

假设我们希望实现以下显示结果：

图片 6-48
启用文本字段的复选框



这里有两个控件：

- 一个带标签的复选框，该标签也是文字字段的标签
- 文本字段本身

开始部分是对这两个控件进行常规声明：

```
<CheckBoxControl property="boolean3" label="Check box 3"/>
<TextBoxControl property="string3" label="String 3"/>
```

这会生成如下所示的面板：

图片 6-49
复选框和文本字段位于单独的行中



首先，我们希望禁止显示文本字段标签 `String 3`。这可通过将文本字段控件的 `showLabel` 属性设置为 `false` 来实现：

```
<CheckBoxControl property="boolean3" label="Check box 3"/>
<TextBoxControl property="string3" label="String 3" showLabel="false"/>
```

文本字段将扩展以填充标签先前占用的区域：

图片 6-50
复选框和不带标签的文本字段



现在，我们希望在同一行显示文本字段和复选框。为此，我们在 `CheckBoxControl` 元素内添加一个 `Layout` 元素，以将行增量设置为 0（默认情况下，每一个控件的行增量为 1）：

```
<CheckBoxControl property="boolean3" label="Check box 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false"/>
```

但是，所得的显示结果如下所示：

图片 6-51
文本字段覆盖了复选框



文本字段向上移动了一行，但它仍然占用了整个行，因此覆盖了复选框。

注：如果显示结果如下所示：

图片 6-52
复选框覆盖了文本字段



由于是先绘制文本字段，然后才绘制复选框，因此文本字段的前几个字符被覆盖了。

无论先绘制哪一个对象，将多个 UI 组件分配到同一个单元格都会得到不理想的行为或未定义的行为，应加以避免。要解决此问题，我们需要在 `TextBoxControl` 元素内添加第二个 `Layout` 元素，以强制从第二个显示列开始显示文本字段：

```
<CheckBoxControl property="boolean3" label="Check box 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1"/>
</TextBoxControl>
```

但是，该解决方案仅解决了部分问题，其显示结果如下所示：

图片 6-53
位置正确但文本字段非常短



两个控件的位置放置正确，但文本字段太短。问题是自定义布局与控件关联后，会覆盖与每种类型的控件相关联的“智能”默认值。在此情况下，`Layout` 元素的默认填充行为（即组件填充其可用单元格的方式）将不填充可用的单元格，而是占用尽可能少的屏幕空间。为改变此情况，我们只需让文本字段去填充水平空间：

```
<CheckBoxControl property="boolean3" label="Check box 3">
    <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
    <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

需要增加一个较小的 `columnWeight` 值以使 Java 正确分配填充的空间。

这样我们会得到预期的布局：

图片 6-54
启用文本字段的复选框



以上布局表面看上去没有错误，但仍然有一个问题需要解决。当前，复选框试图占用整个行，即使我们现在在同一行中放置了另外一个控件。当前看不到该问题，因为复选框标签相对较短，面板上的其他标签（插图中未显示）已经移出了第二个显示列，因此不存在重叠现象。如果将复选框标签变长，该问题就会变得很明显：

```
<CheckBoxControl property="boolean3" label="带有比我们拥有的标签还长的复选框 3">
    <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
    <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

这样做会得到以下显示结果：

图片 6-55
文本字段覆盖长复选框标签



我们唯一需要做的是让复选框将其可用的宽度限制到单一列中：

```
<CheckBoxControl property="boolean3" label="Check box 3 with a much longer label than we had">
    <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
    <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

这样就可以得到我们最终想要的结果：

图片 6-56
长复选框标签完整显示



示例：单选按钮组和文本字段

本示例演示将单选按钮组中的每个按钮与其自身的文本字段相关联的方式。

我们想定义一个如下所示的面板：

图片 6-57
具有文本字段的单选按钮组



这次我们有四个控件：

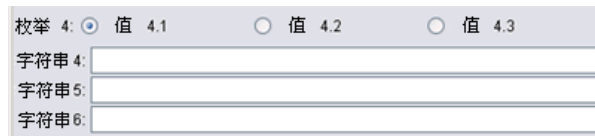
- 一个单选按钮组，用于一个包含三个值的枚举列表
- 三个文本字段，每个文本字段对应一个值

与前面的示例一样，我们首先对这些控件进行简单声明：

```
<RadioButtonGroupControl property="enum4" label="Enum 4"/>
<TextBoxControl property="string4" label="String 4"/>
<TextBoxControl property="string5" label="String 5"/>
<TextBoxControl property="string6" label="String 6"/>
```

这样做将得到以下显示结果：

图片 6-58
具有文本字段和标签的单选按钮组



我们希望使用单选按钮标签标识文本字段，因此我们的第一个任务是在一个具有三行的单列中排列单选按钮，并隐藏文本字段标签：

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3"/>
<TextBoxControl property="string4" label="String 4" showLabel="false"/>
<TextBoxControl property="string5" label="String 5" showLabel="false"/>
<TextBoxControl property="string6" label="String 6" showLabel="false"/>
```

显示结果如下所示：

图片 6-59
单选按钮和文本字段位于同一列中



这里我们已经可以看到一个小小的问题，即单选按钮组标签未与第一个单选按钮对齐。稍后我们将修复此问题，现在我们需要使文本字段与每个对应的单选按钮大致位于一行。

操作步骤与我们在示例 1 中执行的操作类似。我们需要执行以下操作：

- 将单选按钮组的行增量更改为 0。
- 限制网格宽度，使文本字段与单选按钮不重叠。
- 将每个文本字段与其对应的单选按钮放置到同一行。

因此，就像在上一示例中执行的操作一样，我们添加一些 **Layout** 元素。在本示例中，我们按如下所示更改规范文件：

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
  <Layout rowIncrement="0" gridWidth="1" />
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

遗憾的是，我们现在得到了如下所示的显示结果：

图片 6-60
文本字段覆盖了单选按钮



如果我们使用与示例 1 完全相同的 **Layout** 元素，会得到什么样的结果呢？

答案是，与上一示例中的复选框控件不同，单选按钮组（与大多数控件一样）有单独的标签和实际控件。这意味着单选按钮组需要额外的列，因此我们需要从后面的列（第 2 列而不是第 1 列）开始放置文本字段。这样在文本字段的 **Layout** 元素中，我们将 **gridColumn** 的值设置为 2：

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
  <Layout rowIncrement="0" gridWidth="1" />
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```


请注意，虽然我们将文本字段网格列增加到了 2，但我们并没有增加单选按钮组的网格宽度 1。这是由于对于属性控件，大多数 `Layout` 属性仅影响组成控件可编辑区域的 UI 组件，而不影响控件标签。

现在显示结果如下所示：

图片 6-61
文本字段不再覆盖单选按钮



这已经非常接近于我们想要的结果。但是，单选按钮和文本字段之间仍然存在一些对齐问题。

主要问题是单选按钮放置到了单独的子面板中，因此单选按钮与其文本字段之间不存在实际的布局关系。我们唯一需要做的是停止单选按钮组使用子面板：

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3"
useSubPanel="false">
  <Layout rowIncrement="0" gridWidth="1" />
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

最后，我们得到了所需的布局：

图片 6-62
具有文本字段的单选按钮组



使用 `Enabled` 元素控制显示特征

通常可以使用 `Enabled` 元素依据是否满足特定的条件来启用或禁用控件。

面板和属性控件可以拥有与其相关联的条件，以确定各种显示特征。例如，复选框可用于启用关联的文本字段，单选按钮可使另一组隐藏的字段变成可见。

用户界面中的条件通常取决于另一控件的值，而非属性的值。基于属性的条件仅在所作的更改应用到基础对象（例如节点、模型输出或文档输出）时才起作用。在用户界面中，控件需要在相关控件发生变化后立即启用。

格式

```
<Enabled>
  <Condition .../>
  <And .../>
  <Or .../>
  <Not .../>
</Enabled>
```

`Condition` 元素指定要测试的以确定是否启用控件的条件。

通过 `And`、`Or` 和 `Not` 元素可以指定复合条件。

有关详细信息，请参阅第 80 页码第 4 章中的条件。

示例：使用简单条件启用控件

在示例：启用文本字段的复选框第 168 页码中，我们开发了一个复选框，选择该框时可以启用一个文本字段：

图片 6-63
长复选框标签完整显示



我们希望在复选框被选中后立即启用文本字段，而不是在基础对象的属性发生变化时启用文本字段。为此，我们需要添加一个 `Enabled` 条件：

```
<CheckBoxControl property="boolean3" label="Check box 3 with a much longer label than we had">
  <Layout rowIncrement="0" gridWidth="1" />
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
  <Enabled>
    <Condition control="boolean3" op="equals" value="true"/>
  </Enabled>
</TextBoxControl>
```

这可确保文本字段仅在与复选框关联的布尔值为真时才启用。

示例：使用复杂条件启用控件

为了说明复杂条件的编码，我们将观察用 CLEF 开发的广义线性节点的一个对话框选项卡。

该节点对话框有一个“专家”选项卡，其中包含专为用户设计的选项，选项里是这类模型的详细信息。该选项卡上的所有选项最初都是禁用的：

图片 6-64
所有选项均禁用的“专家”选项卡



将模式复选框设置为专家可启用其中一些选项:

图片 6-65
“模式”设置为“专家”的“专家”选项卡



但是，仍有些选项被禁用，例如对话框底部的迭代控件。只有同时满足以下两个条件时才启用此控件：

- 分布设置为正态
- 连接函数设置为恒等式

此组合是“专家”模式中此选项卡的实际默认设置，更改任意这两个组合框的设置都会启用迭代：

图片 6-66
更改“分布”设置可启用“迭代”按钮



实现上述目的的代码包含在迭代按钮的 `PropertiesSubPanel` 声明中，如下所示：

```
<PropertiesSubPanel buttonLabel="Iterations..." buttonLabelKey= ...
  <Enabled>
    <And>
      <Condition control="mode" op="equals" value="Expert"/>
      <Not>
        <And>
          <Condition control="distribution" op="equals" value="NORMAL"/>
          <Condition control="link_function" op="equals" value="IDENTITY"/>
        </And>
      </Not>
    </And>
  </Enabled>
  ...
```

```
</PropertiesSubPanel>
```

外层 **And** 部分中的 **Condition** 元素指定在进行任何更改之前必须将模式设置为专家。假设满足此条件，则 **Not** 部分指定仅在同时满足内层 **And** 部分的两个条件时才禁用此按钮。因此，在“专家”模式中，如果分布或连接函数的设置值与其默认值不同，则启用迭代。

使用 Visible 元素控制显示特征

还可以根据指定的情况使用条件显示或隐藏控件。通过 **Visible** 元素可实现上述目的。

格式

```
<Visible>
  <Condition .../>
  <And .../>
  <Or .../>
  <Not .../>
</Visible>
```

Condition 元素指定要测试的以确定是否显示控件的条件。

通过 **And**、**Or** 和 **Not** 元素可以指定复合条件。

有关详细信息，请参阅第 80 页码第 4 章中的条件。

示例

下面的示例演示仅在满足 **source_language** 条件时才显示指定的属性面板：

```
<PropertiesPanel>
  <Visible>
    <Condition control="source_language" op="equals" value="eng" />
  </Visible>
  ...
</PropertiesPanel>
```

自定义输出窗口

对于模型输出、文档输出和交互输出对象（而非节点），扩展时可以使用自定义窗口完全代替默认的输出窗口。这是作为标准的 **java.awt.Frame** 类实现的。

要提供自定义窗口，请指定一个 Java 类作为 **UserInterface** 元素的 **frameClass** 属性，如下所示：

```
<DocumentOutput id="my_modelling_node" type="modelBuilder" ...>
  <Properties>
    <Property name="use_custom_type" valueType="boolean" .../>
    ...
  </Properties>
</DocumentOutput>
```

```
</Properties>
<UserInterface frameClass=" com.myextension.MyOutputFrame" />
...
</DocumentOutput>
```

所指定的类必须实现通过 CLEF 客户端 API 定义的 `ExtensionObjectFrame` 接口。这包含窗口的生命周期：

- 访问基础 `java.awt.Frame`
- 初始化窗口，包括访问输出对象和会话
- 在要保存或删除对象时同步窗口和基础对象
- 窗口处理

有关详细信息，请参阅第 193 页码第 9 章中的客户端 API 类。

添加帮助系统

帮助系统的类型

开发 CLEF 扩展时，通常希望包含联机帮助系统以说明如何使用该扩展。CLEF 支持下列类型的帮助系统：

- HTML 帮助
- JavaHelp

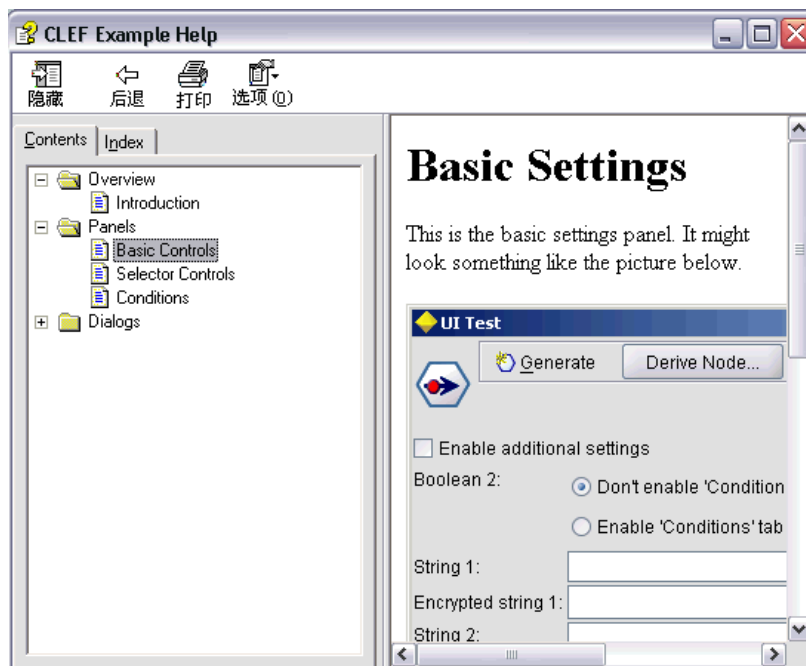
HTML 帮助

HTML 帮助是 Microsoft 开发的专有格式，只能在 Windows 平台上运行。HTML 帮助系统由单个 `.htm` 或 `.html` 文件编译成压缩格式，形成扩展名为 `.chm` 的独立文件。IBM® SPSS® Modeler 自己的帮助系统以 HTML 帮助格式提供。

HTML 帮助支持目录、索引和全文本搜索功能（术语表可通过弹出窗口的形式实现）。可以使用 HTML 编辑器或商用帮助创作工具创建 `.htm` 或 `.html` 的源主题文件。要生成 `.chm` 文件，可以选择使用 HTML Help Workshop，该工具可从“Microsoft 下载中心”网站免费下载（有关生成 `.chm` 文件的详细信息，请参见 HTML Help Workshop 的帮助系统）。也可以使用支持 HTML 帮助格式的帮助创作工具将所用的主题文件和所有图形文件编译为 `.chm` 文件。

HTML 帮助浏览器的外观如下所示：

图片 7-1
HTML 帮助浏览器



JavaHelp

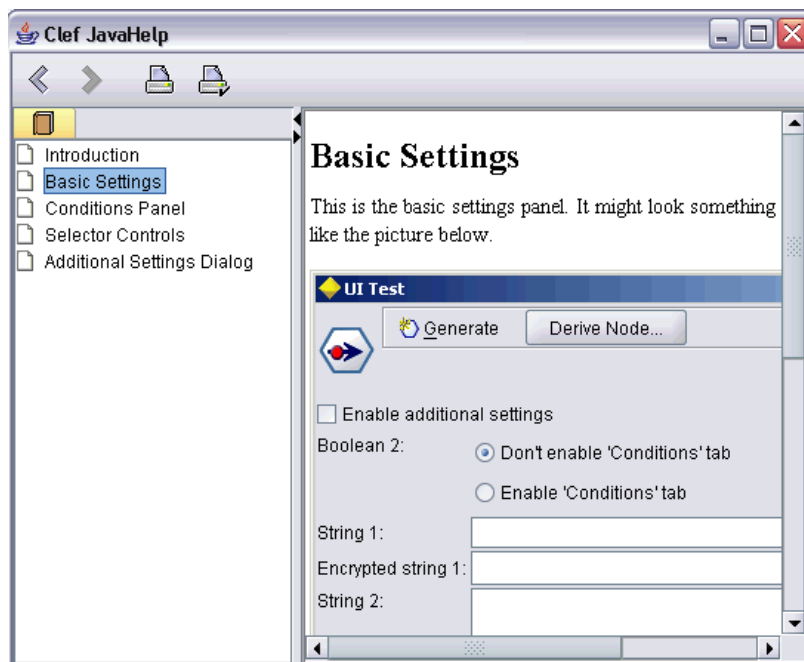
JavaHelp 是一种开放源代码帮助格式，由 Sun Microsystems 开发，可在支持 Java 的所有平台上运行。JavaHelp 系统中包含以下文件：

- .htm 或 .html 的源主题文件
- 主题中使用的所有图形文件
- 控制帮助系统的帮助集文件（扩展名为 .hs）
- map.xml 文件，用于将主题 ID 和主题文件相关联，并定义显示帮助主题的窗口
- index.xml 文件，其中包含索引条目
- toc.xml 文件，其中包含目录条目

JavaHelp 支持目录、索引、全文本搜索和术语表功能。可以使用 HTML 编辑器或商用帮助创作工具创建 .htm 或 .html 的源文件。您将需要 JavaHelp 软件，可从 Sun Developer Network 网站免费下载该工具（有关详细信息，请参见 JavaHelp System User's Guide，此文档也可以从该网站获取。）

JavaHelp 浏览器的外观如下所示：

图片 7-2
JavaHelp 浏览器



实施帮助系统

本节介绍如何在规范文件中定义帮助系统的相关组件。

定义帮助系统的位置和类型

用于扩展的帮助系统类型（如果有）在扩展的规范文件“资源”部分的 `HelpInfo` 元素中定义。

格式

```
<Resources>
...
  <HelpInfo id="名称" type="help_type" path="help_path" helpset="helpset_loc"
    default="default_topicID" />
...
</Resources>
```

其中：

`id`（必需）是该扩展的帮助信息的标识符。

`type`（必需）表示帮助类型，是下列内容之一：

- `htmlhelp`—HTML 帮助，包含在由 `path` 属性标识的 `.chm` 文件中。
- `javahelp` - JavaHelp，使用帮助集 (`.hs`) 文件（由 `helpset` 属性标识）和帮助源文件及相关文件。

如果帮助类型为 `htmlhelp`，则需要下列附加属性：

- `path` - 包含帮助系统的 `.chm` 文件的位置（相对于规范文件）和名称。

如果帮助类型为 `javahelp`，则需要下列属性：

- `helpset` - 要使用的 `.hs` 帮助集文件的位置（相对于规范文件）和名称。
- `default` - 在没有为特定的选项卡指定主题时，要显示的默认主题的标识符。

如果没有指定 `HelpInfo` 元素，则没有帮助与该扩展关联。

示例

第一个示例说明 HTML 帮助的 `HelpInfo` 元素：

```
<HelpInfo id="help" type="htmlhelp" path="help/mynode.chm" />
```

JavaHelp 系统的等价元素是：

```
<HelpInfo id="help" type="javahelp" helpset="help/mynode.hs"/>
```

请注意，如果使用 JavaHelp，关联文件（图像、映射文件、索引和内容文件）必须与 `.hs` 帮助集文件位于同一文件夹中。

指定要显示的特定帮助主题

可以指定用户在节点对话框、特定选项卡或属性子面板中调用帮助时要显示的特定帮助主题。要实现该功能，可使用节点、选项卡或属性子面板规范的 `helpLink` 属性。

如果没有指定 `helpLink` 属性，在用户调用帮助时将显示帮助系统的默认主题。

有关详细信息，请参见 [节点第 54 页码](#)、[Tabs 第 124 页码](#) 和 [属性子面板第 138 页码](#) 中对 `helpLink` 属性的介绍。

示例

本示例假设您要使用 HTML 帮助，介绍了如何操作才能显示不同的上下文主题，具体取决于用户选择帮助时选中的窗口。

```
<Resources>
  ...
  <HelpInfo id="help" type="htmlhelp" path="help/mynode.chm"/>
  ...
</Resources>
...
<Node id="mynode" scriptName="my_node" type="dataTransformer" palette="recordOp"
  label="Sorter" description="Sorts a data file" >
```

```

...
<Tabs defaultTab="1">
  <Tab label="Basic Controls" labelKey="basicControlsTab.LABEL"
    helpLink="basic_controls.htm">
    <PropertiesPanel>
      ...
      <PropertiesSubPanel buttonLabel="Additional settings..."
        buttonLabelKey="AdditionalOptions.LABEL" dialogTitle="Additional Settings"
        dialogTitleKey="AdditionalOptionsDialog.LABEL" helpLink="addsettingsdlg.htm">
      ...
    </Tab>
    <Tab label="Selector Controls" labelKey="selectorControlsTab.LABEL"
      helpLink="selector_controls.htm">
    ...
  </Tab>
  ...
</Node>

```

这指定了：如果焦点位于“基本控制”选项卡时用户选择帮助，则显示 `mynode.chm` 帮助文件中 `basic_controls.htm` 中的主题。如果用户随后单击其他设置按钮打开“其他设置”对话框，并选择该对话框中的帮助，将显示 `addsettingsdlg.htm` 中的主题。如果用户随后关闭“其他设置”对话框，而选择“选择器控件”选项卡，并再次选择帮助，将显示 `selector_controls.htm` 中的主题。

对于 JavaHelp，`helpLink` 属性的值必须与 `map.xml` 文件中 `target` 属性的值相匹配。例如，如果 `map.xml` 文件包含下列内容：

```

<map version="1.0">
  ...
  <mapID target="basic_controls" url="basic_controls.htm"/>
  ...
</map>

```

必须为相应的 `helpLink` 属性赋予下列值：

```
helpLink="basic_controls"
```

这是因为当 JavaHelp 被调用时，它将读取 `target` 属性的值并将其映射到相关的 `url` 值，以找到要显示的正确文件。

本地化和辅助功能

简介

本地化指调整软件、帮助和文档，使它们适应特定区域的过程。包括以相应区域的语言翻译用户界面、帮助和文档以及测试系统。如果您要将扩展部分分发给非本地区域的用户，应该分发扩展部分的本地化版本。

辅助功能，这里是指在用户界面中包含的功能，可使身体有残疾（例如视觉障碍或手的灵活度有限）的用户能够更方便地访问系统。

本地化

IBM® SPSS® Modeler 自身已针对全球很多地区进行了本地化。对于所有受支持的语言版本，如果用户将 Windows 区域选项设置为自己所在的区域，则标准 IBM® SPSS® Modeler UI 组件将以相应的语言显示，例如：

- 系统菜单和菜单条目
- 系统按钮（“生成”、“确定”、“执行”、“取消”、“应用”和“重置”）
- 标准对话框选项卡（如果使用，则使用“注解”和“调试”）
- 错误和系统消息（例如“尚未保存该对象”。）

在扩展部分使用这些标准 SPSS Modeler 组件的位置，这些组件将自动以选定的语言（如果该语言受支持）显示。

对于扩展部分的其他组件，CLEF 提供了有助于本地化的功能。可以本地化的内容包括：

- 节点名称（在选项板和工作区上）
- 模型名称（在管理器窗格的“模型”选项卡上）
- 文档名称（在管理器窗格的“输出”选项卡上）
- 与操作相关的图标图像的位置
- 工具提示文本
- 帮助系统
- 节点对话框：
 - 标题栏文本
 - 自定义菜单和菜单条目
 - 字段、属性、按钮和选项卡标签
 - 静态文本
- 系统和错误消息

在翻译项目时，文本字符串应尽量缩短，以便能够保留较长的文本。

系统和错误消息可以通过规范文件、属性文件和服务器端 API 的组合进行本地化。有关详细信息，请参阅第 212 页码第 9 章中的状态详细信息文档。

属性文件

可以本地化的项目的文本字符串存储在作为**属性文件**的文件中，该文件使用标准 Java 格式存储本地化资源数据包。每个属性文件都包含一连串的记录，每个记录分别对应于扩展部分每个本地化后的项目。每个记录中的字段都与规范文件中的 `labelKey` 属性相对应，从而使 CLEF 可以从属性文件中读取相应的文本字符串，并将其显示在正确的位置。

属性文件的扩展名必须是 `.properties`，且该文件必须与其相关节点的规范文件位于同一目录中。IBM® SPSS® Modeler 开始会查找默认的属性文件，名为：

```
path.properties
```

其中 `path` 是定义了属性数据包的 `Bundle` 元素（在“资源”部分）中的 `path` 属性。例如：

```
<Bundle id="bundle" path="my_resources"/>
```

如果没有默认的属性文件，SPSS Modeler 会从规范文件中的定义读取文本字符串。

必须有本地化支持的各种语言的属性文件。非默认语言的文件按照文件名的后缀加以区分。例如：

```
my_resources.properties  
my_resources_de.properties  
my_resources_fr.properties
```

后缀符合语言代码的双字符 ISO 639-1 标准。

属性文件中的每个记录具有如下格式：

```
id=text_string
```

其中：

`id` 是规范文件中 `buttonLabelKey`、`descriptionKey`、`dialogTitleKey`、`falseLabelKey`、`imagePathKey`、`labelKey`、`messageKey`、`textKey` 或 `trueLabelKey` 属性的标识符。为便于区分，该标识符的后缀通常是 `.LABEL`。当然，它可以有任何后缀或没有后缀，具体取决于它在规范文件中的出现形式。

`text_string` 是项目的文本。

示例：本地化对话框选项卡

本节点对话框中本地化选项卡示例使用两个属性文件，即默认（英语）版本和法语版本，位置如下：

```
extension_folder\my_resources.properties  
extension_folder\my_resources_fr.properties
```

其中，`extension_folder` 指包含规范文件的文件夹。

在规范文件中，属性文件通过 `Resources` 部分中的 `Bundle` 元素引用：

```
<Resources>
  <Bundle id="bundle" type="properties" path="my_resources"/>
</Resources>
```

请注意，`path` 属性一定不能包含语言扩展名或 `.properties` 后缀。

规范文件的其他相关部分是：

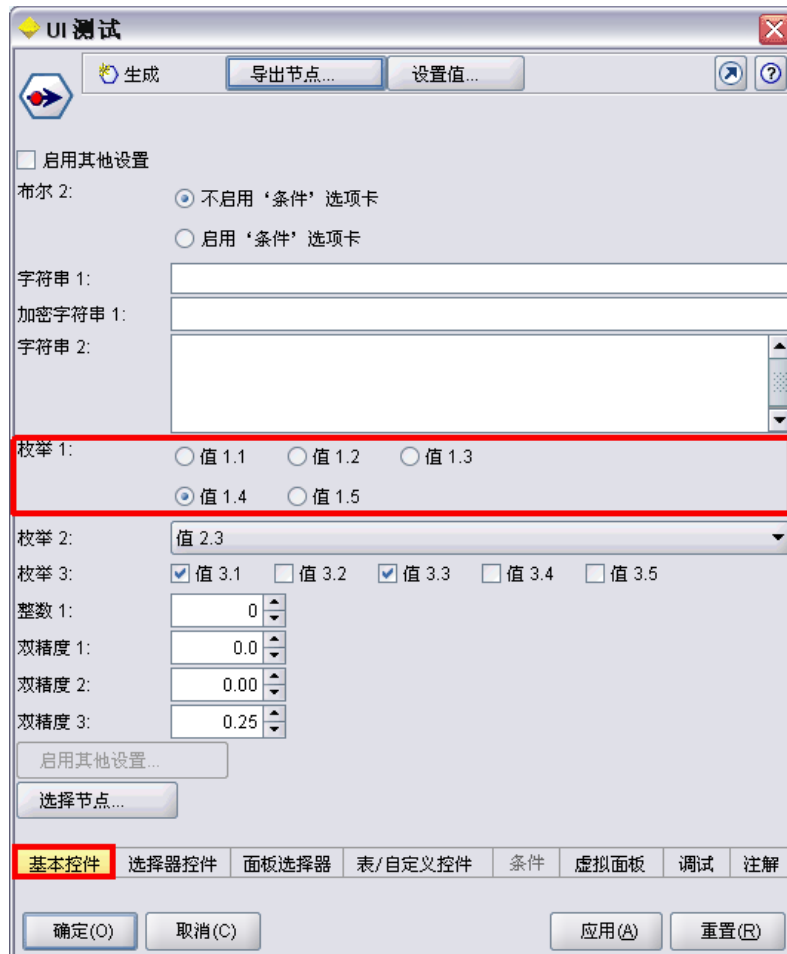
```
<Node id="uiTest" scriptName="ui_test" type="dataTransformer" palette="record0p" label="UI Test" ...>
  <Properties>
    <Property name="enum1" valueType="enum" defaultValue="value4">
      <Enumeration>
        <Enum value="value1" label="Value 1.1" labelKey="enum1.value1.LABEL"/>
        <Enum value="value2" label="Value 1.2" labelKey="enum1.value2.LABEL"/>
        <Enum value="value3" label="Value 1.3" labelKey="enum1.value3.LABEL"/>
        <Enum value="value4" label="Value 1.4" labelKey="enum1.value4.LABEL"/>
        <Enum value="value5" label="Value 1.5" labelKey="enum1.value5.LABEL"/>
      </Enumeration>
    </Property>
  </Properties>
  ...
  <UserInterface ...>
    <Tabs defaultTab="1">
      <Tab label="Basic Controls" labelKey="basicControlsTab.LABEL" ... >
        ...
      </UserInterface>
    ...
  </Node>
```

在属性文件中，英语版属性文件包括以下记录：

```
basicControlsTab.LABEL=Basic Controls
enum1.value1.LABEL=Value 1.1
enum1.value2.LABEL=Value 1.2
enum1.value3.LABEL=Value 1.3
enum1.value4.LABEL=Value 1.4
enum1.value5.LABEL=Value 1.5
```

此处突出显示了受这些记录影响的对话框部分：

图片 8-1
不可本地化的选项卡

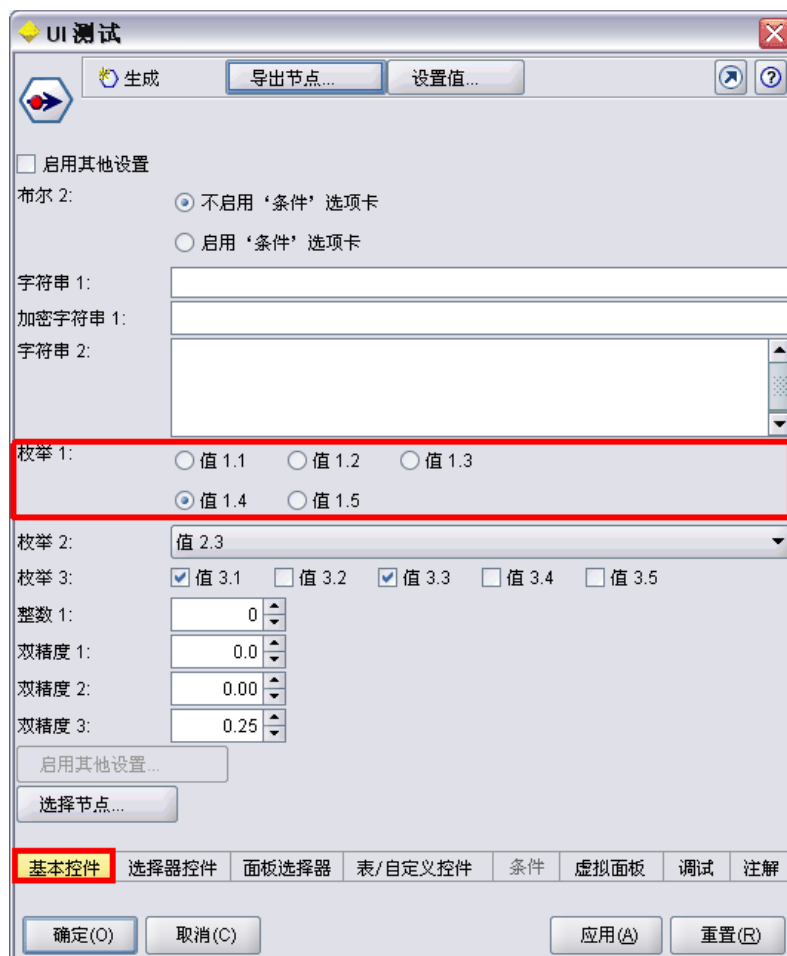


法语版属性文件（my_resources_fr.properties）的相应部分是：

```
basicControlsTab.LABEL=Contrles de Base
enum1.value1.LABEL=Valeur 1,1
enum1.value2.LABEL=Valeur 1,2
enum1.value3.LABEL=Valeur 1,3
enum1.value4.LABEL=Valeur 1,4
enum1.value5.LABEL=Valeur 1,5
```

这些记录将导致该屏幕的相关部分显示翻译后的文本：

图片 8-2
本地化后的选项卡



请注意，屏幕底部四个按钮的本地化由 IBM® SPSS® Modeler 功能（而不是 CLEF）处理。

示例：使用特殊字符

在属性文件中，您将需要对标准 ASCII 文本字符串的所有特殊字符使用 Unicode 转义序列。下面是已本地化为法语的部分属性文件：

```
Genlinnode.LABEL=Lin\u00e9aire g\u00e9n\u00e9ralis\u00e9
```

```
Fields.LABEL=Champs
```

```
Model.LABEL=Mod\u00e8le
```

```
Expert.LABEL=Expert
```

```
inputFields.LABEL=Entr\u00e9es
```

```
targetField.LABEL=Cible
```

```
...
```

对于使用非拉丁语字符的语言（例如日语或汉语），需要对整个文本字符串使用 Unicode 转义序列。此处是针对日语本地化后的相同记录集：

```
GenI nnode. LABEL=\u4e00\u822c\u5316\u7dda\u578b  
  
Fie lds. LABEL=\u30d5\u30a3\u30fc\u30eb\u30c9  
Model. LABEL=\u30e2\u30c7\u30eb  
Expert. LABEL=\u30a8\u30ad\u30b9\u30d1\u30fc\u30c8  
  
inputFields. LABEL=\u5165\u529b  
targetField. LABEL=\u5bfe\u8c61  
...
```

帮助文件

如果要本地化具有帮助系统的扩展部分，应当也提供本地化后的帮助系统版本。应当针对每个本地化的扩展部分提供一个本地化帮助系统。

本地化 HTML 帮助

如果您要本地化的扩展部分使用 HTML 帮助文件（后缀为 `.chm`），可以使用本地化后的版本替换默认的 `.chm` 文件。有关 HTML 帮助系统的详细信息，请参阅[HTML 帮助第 180 页码](#)。

要创建本地化后的 `.chm` 文件：

- ▶ 创建构成帮助系统的各个帮助主题（`.htm` 或 `.html`）文件的翻译版本，同时保持文件名不变。
- ▶ 如果需要，可以使用包含在帮助系统中的本地化版本的图形（例如，屏幕快照）。
- ▶ 使用 Microsoft HTML Help Workshop 或其他帮助创作工具将上述文件编译为本地化后的 `.chm` 文件。
- ▶ 测试帮助系统中本地化后的节点。 [有关详细信息，请参阅第 191 页码测试本地化后的 CLEF 节点。](#)

本地化 JavaHelp

如果要本地化的扩展部分使用 JavaHelp 系统，将需要针对各种受支持的语言提供帮助源文件的本地化版本。JavaHelp 负责显示正确的本地化版本（如果存在）。 [有关详细信息，请参阅第 181 页码第 7 章中的 JavaHelp。](#)

要创建本地化后的 JavaHelp 系统：

- ▶ 创建构成帮助系统的各个帮助主题（`.htm` 或 `.html`）文件的翻译版本，同时保持文件名不变。
- ▶ 如果需要，可以使用包含在帮助系统中的本地化版本的图形（例如，屏幕快照）。
- ▶ 生成帮助集和其他所需文件（映射文件、内容和索引文件）。

- ▶ 测试帮助系统中本地化后的节点。 [有关详细信息，请参阅第 191 页码测试本地化后的 CLEF 节点。](#)

测试本地化后的 CLEF 节点

要测试本地化后的节点及其帮助系统：

- ▶ 在本地化节点的规范文件中的“资源”部分，更改 `HelpInfo` 元素的 `path` 属性以引用本地化后的 `.chm` 或 `.hs` 文件。例如，对于 HTML 帮助，您可以使用：

```
<Resources>
...
  <HelpInfo id="help" type="HTMLHelp" path="help/mynode_fr.chm "/>
</Resources>
```

对于 JavaHelp，可以使用：

```
<Resources>
...
  <HelpInfo id="help" type="javahelp" helpset="help/mynode_fr.hs "/>
</Resources>
```

- ▶ 将本地化后的 `.chm` 或 `.jar` 文件复制到 `path` 属性标明的位置。
- ▶ 设置 Windows 区域的所需区域：
控制面板 > 区域和语言选项 > 区域选项 > 标准和格式 > <语言>
- ▶ 启动 IBM® SPSS® Modeler，确保它以所需的语言显示。
- ▶ 将本地化后的节点添加到 SPSS Modeler。 [有关详细信息，请参阅第 219 页码第 10 章中的测试 CLEF 扩展。](#)
- ▶ 将节点副本置于工作区。
打开节点对话框，检查它是否以所需的语言正常显示。
- ▶ 单击对话框中的“帮助”按钮，确保以所需语言显示正确的帮助主题。

辅助功能

所有标准的 IBM® SPSS® Modeler 辅助功能（例如，与鼠标操作等效的键盘操作和屏幕读取器支持）有利于 CLEF 节点。 [有关详细信息，请参阅附录 A 中的 IBM SPSS Modeler 辅助功能概述中的 IBM SPSS Modeler 15 用户指南。](#)

另外，出于提供辅助功能的考虑，可以提供带自定义工具提示文本的 CLEF 节点。

您也可以指定键盘组合为最终用户提供访问您在 CLEF 中添加的各个用户界面功能的替代访问方式。 [有关详细信息，请参阅第 126 页码第 6 章中的访问键和键盘快捷键。](#)

对于操作按钮和被分类为控制器的各个屏幕组件（例如复选框或单选按钮组），可以定义：

- 标签
- 描述

标签是在屏幕上作为组件的名称显示的文本，可用屏幕读取器软件读取。对于有视力障碍的用户，可通过“用户选项”对话框“显示”选项卡上的控件更改标签的显示字体大小，该对话框的访问方法是：

工具(S) > 用户选项

说明是鼠标指针悬停在组件上时显示的工具提示文本。工具提示为只能由名称传达信息的组件提供了更多信息。工具提示也只能由配置为读取它们的屏幕读取器读取。

标签和说明通过元素（在规范文件中定义组件）中的 `label` 和 `description` 属性定义。这两项分别可以通过 `labelKey` 和 `descriptionKey` 属性进行本地化。

示例

该操作按钮示例介绍了标签和说明功能的用途。

```
<Action id="setValue" label="Set Value..." labelKey="setValue.LABEL"
  description="Sets a value" descriptionKey="setValue.TOOLTIP"/>
```

程序设计

关于 CLEF 节点的程序设计

为了使节点能够执行无法在规范文件中定义的处理，CLEF 提供了下列应用程序编程接口 (API)，您的程序可以调用这些接口：

- **客户端 API**。一种 Java API，可供需要规范文件中没有直接提供的其他控件、用户界面组件或交互的扩展使用。
- **Predictive Server API (PSAPI)**。一种 Java API，公开可供需要数据挖掘和预测分析功能的应用程序使用的 IBM® SPSS® Modeler 功能。PSAPI 和 SPSS Modeler 数据挖掘平台共享相同的底层技术。
- **服务器端 API**。一种基于 C 的 API，它涵盖许多方面，如设置和获取执行设置、这些设置的持续性、执行反馈、作业控制（如中断执行）、SQL 生成和返回的对象。

CLEF API 文档

以下各节提供了客户端和服务端 API 的概述。更完整的 API 文档以 zip 文件的形式包含在 IBM® SPSS® Modeler 安装中，必须提取后才能使用。

要提取 API 文档：

- ▶ 在产品 DVD 的 \Documentation\en 文件夹中找到文件 clef_apidoc.zip。
- ▶ 使用 WinZip 或类似工具，将 zip 文件内容提取到任何适当目录中。这样做会在该目录中创建一个包含所有 API 文档的 clef_apidoc 子文件夹。

要查看 API 文档：

- ▶ 定位到 clef_apidoc 子文件夹，然后打开 clef_apidoc.htm 文件。
- ▶ 根据需要选择 PSAPI/客户端或服务端选项。

客户端 API

CLEF 提供了多个包含可用于进行客户端处理的方法的 Java 类。例如，通过 **DataModelProvider** 类，可以计算由于对输入数据模型的更改太复杂而无法使用规范文件提供的功能的输出数据模型。

客户端 API 类

客户端类如下。

表 9-1
客户端 API 类

Class	描述
ActionHandler	允许扩展管理用户通过菜单选项和工具栏按钮请求的操作
DataModelProvider	允许对数据模型进行复杂更改的节点使用 Java 计算输出数据模型
ExtensionObjectFrame	定义与用于显示模型或文档输出的窗口相关联的功能
ExtensionObjectPanel	定义与扩展对象面板相关联的功能
PropertyControl	定义与属性面板中的自定义属性控件相关联的功能

客户端 API 文档中提供了这些类的全部详细信息。 [有关详细信息，请参阅第 193 页码 CLEF API 文档。](#)

使用客户端 API

要将客户端函数调用包含到 CLEF 节点中：

- ▶ 创建包含函数调用的 .java 源文件。
- ▶ 将源文件编译到 .class 文件中。
- ▶ 可以将多个 .class 文件合并到一个 .jar 文件中，然后在规范文件中包含对该 .jar 文件的引用—例如：

```
<Resources>
...
<JarFile id="selfjar" path="selflearning.jar"/>
...
</Resources>
```

某些 CLEF 元素允许您显式引用某一个类。例如，可以在规范文件的 PropertyControl 元素的 controlClass 属性中包含一个类引用：

```
<PropertyControl property="target_field_values_specify" ...
  controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" ... />
```

其中，CustomListControl 是实现属性控件的类的名称；com.spss.clef.selflearning.propertycontrols.list 是该类在 JarFile 元素中声明的 .jar 文件内的路径。

[有关详细信息，请参阅第 38 页码第 4 章中的资源部分。](#)

您还会发现看一看与此版本一同提供的示例节点的源代码也会很有用。 [有关详细信息，请参阅第 33 页码第 3 章中的检查源代码。](#)

Predictive Server API (PSAPI)

PSAPI 提供了一种面向底层 Predictive Server 技术的编程接口。PSAPI 的主要元素均被指定为 Java 接口。其中大部分接口均通过由 PSAPI 提供但不构成 PSAPI 规范的一部分的内部类来实现。此方法旨在保护 PSAPI 用户不受 Predictive Server 技术更改（如结构更改、专用客户端/服务器协议更改等）的影响。

PSAPI 文档中提供了这些类的全部详细信息。 [有关详细信息，请参阅第 193 页码 CLEF API 文档。](#)

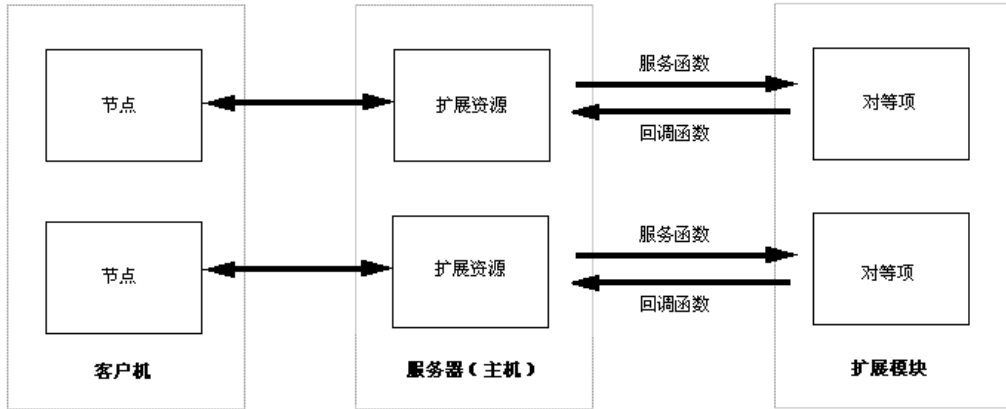
服务器端 API

服务器端 API 定义为 C 语言 API，但仍将支持用 C++ 编写的实现。扩展模块的开发人员可以选择用 C 或 C++ 来对 C 语言 API 进行直接编程。如果开发人员有绑定到 C API 的方法，则也可以使用其他语言。CLEF 还提供了多个 C++ 帮助程序源文件，这些文件相当于某些 C API 的包装。

体系结构

客户端上的扩展节点由服务器上的扩展对等补充。对等由以服务器承载的共享库形式实现的扩展模块定义。节点与其对等之间的通信由服务器管理的扩展资源传递。资源调用由扩展模块定义的服务函数可创建并操作其对等，同时对等使用回调函数可向其主机请求信息和服务。

图片 9-1
CLEF API 体系结构



服务函数

服务函数由扩展模块实现。扩展模块必须实现标记为必需的所有函数，并且可以实现任一或所有未作此标记的函数。

有两种类型的服务函数：

- 模块函数
- 对等函数

以下各节提供了服务函数的概述。有关这些函数的更详细说明可以在服务器端 API 文档中找到，如下所示：

- ▶ 在“CLEF API 文档”屏幕中，选择服务器端 API 概述。
- ▶ 单击模块选项卡。
- ▶ 选择由扩展模块实现的 API 服务函数。

有关访问 CLEF API 文档的信息，请参阅[CLEF API 文档第 193 页码](#)。

模块函数

模块函数从单线程调用。

函数	需要吗?	描述
<code>clmext_initialise</code>	Yes	初始化扩展模块
<code>clmext_cleanup</code>	Yes	释放由扩展模块分配的资源
<code>clmext_getModuleInformation</code>	No	获取有关扩展模块的信息
<code>clmext_create_peer</code>	Yes	为扩展节点创建对等实例
<code>clmext_destroy_peer</code>	Yes	释放对等实例

对等函数

对等函数应用于通过先前调用 `clmext_create_peer` 返回的对等实例句柄。仅当对等句柄唯一时，才可以从各个单独的线程并发调用对等函数。但存在一种例外情况，即可以随时从任何线程调用 `clmext_peer_cancelExecution` 函数（如果已定义）以中断另一线程上运行时间很长的执行。

函数	需要吗?	描述
<code>clmext_peer_configure</code>	Yes	指示对等实例处理其参数和数据模型
<code>clmext_peer_getDataModel</code>	Yes	获取对等实例的输出数据模型
<code>clmext_peer_getCatalogueInformation</code>	No	获取模块的分类信息
<code>clmext_peer_getExecutionRequirements</code>	No	获取对等实例的执行要求
<code>clmext_peer_beginExecution</code>	Yes	开始执行对等实例
<code>clmext_peer_nextRecord</code>	Yes	移到对等结果集中的下一个记录
<code>clmext_peer_getRecordValue</code>	Yes	返回上次提取的结果记录中指定字段的值
<code>clmext_peer_endExecution</code>	Yes	指示执行已完成的对等实例
<code>clmext_peer_cancelExecution</code>	No	从单独的线程调用以中断运行时间很长的 <code>beginExecution</code> 或 <code>nextRecord</code> 函数调用

函数	需要吗?	描述
clemtxt_peer_getSQLGeneration	No	从对等实例生成 SQL
clemtxt_peer_getErrorDetail	Yes	检索对等上有关上一个模块特定的错误的补充信息

下列对等函数设计为与交互模型构建器一起使用：

函数	需要吗?	描述
clemtxt_peer_beginInteraction	No	开始与对等实例交互
clemtxt_peer_request	No	执行对等上的交互请求
clemtxt_peer_getRequestReply	No	检索上一个成功执行的请求的回复
clemtxt_peer_endInteraction	No	指示交互已完成的对等实例

回调函数

当扩展模块需要来自主机进程的信息或服务时，它必须通过**回调**来进行此操作。回调应用于**句柄**，句柄是标识请求目标的指针。

通过在该调用所定向的 IBM® SPSS® Modeler 对象的句柄中进行传递可调用回调。句柄作为服务函数的参数传递到扩展模块中。

如果回调函数失败，它应在相关联的模块特定的错误代码（由 `CLEMEXTErrorCode` 表示）中返回一些更详细的信息。反过来，模块可以通过返回回调错误并传递此详细信息来管理失败的调用函数，以便主机能够检测到它。

可用的回调函数类型有：

- 主机函数
- 节点函数
- 迭代器函数
- 进度函数
- 通道函数（仅适用于交互模型）

以下各节提供了回调函数的概述。有关这些函数的更详细说明可以在服务器端 API 文档中找到，如下所示：

- ▶ 在“CLEF API 文档”屏幕中，选择服务器端 API 概述。
- ▶ 单击模块选项卡。
- ▶ 选择一般回调。

有关访问 CLEF API 文档的信息，请参阅[CLEF API 文档第 193 页码](#)。

主机函数

主机函数定义在通过 `clmext_initialise` 传递的主机句柄中。

函数	描述
<code>clmext_host_getHostInformation</code>	获取有关主机环境的静态信息

节点函数

节点函数定义在通过 `clmext_create_peer` 传递的节点句柄中。

函数	描述
<code>clmext_node_getNodeInformation</code>	获取有关节点的静态信息
<code>clmext_node_getParameters</code>	获取节点的参数
<code>clmext_node_getDataModel</code>	获取节点的输入数据模型
<code>clmext_node_getOutputDataModel</code>	获取节点的输出数据模型
<code>clmext_node_getSQLGeneration</code>	获取节点的上游 SQL 生成信息
<code>clmext_node_getPassword</code>	检索密码标识符的纯文本
<code>clmext_node_getFilePath</code>	检索执行期间在客户端和服务端之间交换的文件的完整路径

迭代器函数

迭代器函数定义在通过 `clmext_peer_beginExecution` 传递的迭代器句柄中。迭代器公开扩展模块的输入数据集。

函数	描述
<code>clmext_iterator_nextRecord</code>	移到输入数据集中的下一个记录
<code>clmext_iterator_getRecordValue</code>	返回上次提取的输入记录中指定字段的值
<code>clmext_iterator_rewind</code>	恢复输入数据集的状态，以便下一次调用 <code>nextRecord</code> 时会移到数据集中的第一个记录

进度函数

进度函数定义在通过 `clmext_peer_beginExecution` 传递的进度句柄中。

函数	描述
<code>clmext_progress_report</code>	向主机汇报进度

通道函数

通道函数只能与交互模型一起使用，并且定义在通过 `clemext_peer_beginInteraction` 传递的通道句柄中。

函数	描述
<code>clemext_channel_send</code>	向客户端发送异步消息，启用一个对等线程以报告进度和状态信息

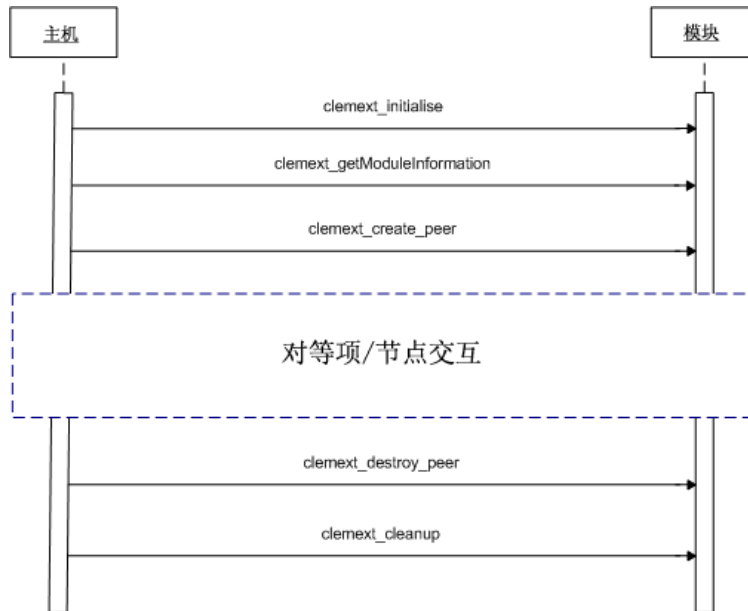
处理流程

扩展模块调用多个服务和回调函数以执行其处理。实际调用的函数取决于模块需要执行的处理。

示例

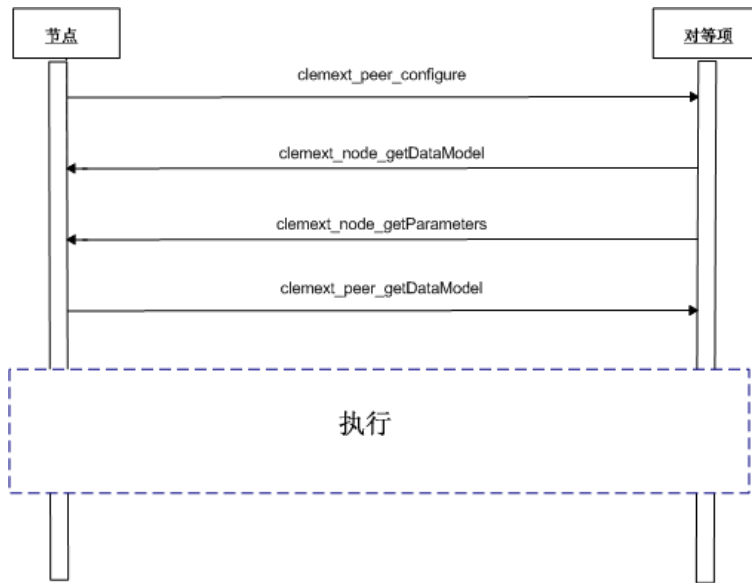
典型模块执行的顺序示意图如下所示：

图片 9-2
典型处理流程



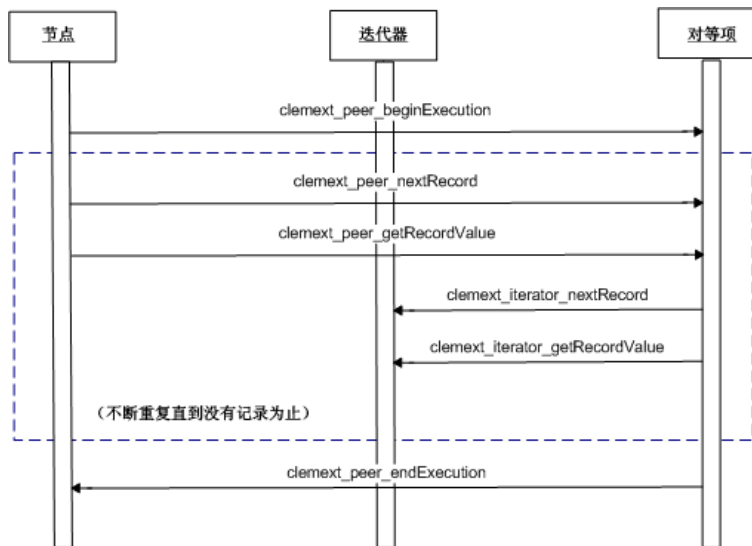
对等/节点交互块如下所示：

图片 9-3
典型对等/节点交互块



典型执行块如下所示：

图片 9-4
典型执行块



注意：

- 模块可能会在服务器启动时加载到 IBM® SPSS® Modeler 服务器进程中，也可能在以后首次需要其服务时根据需要加载。
- 模块加载后，主机调用一次服务函数 `clemext_initialise`。
- 模块加载并初始化后，主机可能会使用服务函数 `clemext_getModuleInformation` 查询该模块。

- 模块加载后，将通过该模块提供的对等对象调用其服务。在模块内，对等对象由服务函数 `clmext_create_peer` 作为主机节点对象的对应对象创建，以按照主机应用程序的指示管理任务的执行。同一类型可以存在多个对等对象，并且可同时在一个进程中并发执行。
- 创建对等对象后，可以通过服务函数 `clmext_peer_configure` 来配置它。
- 此时，对等可以执行回调函数以获取客户端的信息，如 `clmext_node_getDataModel` 和 `clmext_node_getParameters`。
- SPSS Modeler 通过 `clmext_peer_getDataModel` 服务函数来获取对等实例的输出数据模型。
- 可通过 `clmext_peer_beginExecution` 服务函数开始执行对等实例。
- `clmext_peer_nextRecord` 服务函数将焦点移到对等结果集中的下一个记录（如果是首次调用该函数，则移到第一个记录）。然后是 `clmext_peer_getRecordValue` 服务函数，它返回当前记录中指定字段的值。
- 可通过 CLEF 模块调用迭代器回调函数 `clmext_iterator_nextRecord` 和 `clmext_iterator_getRecordValue`，以对输入记录进行排序并返回指定的字段值。
- 可通过调用 `clmext_peer_endExecution` 服务函数来结束对等函数的执行。
- 可通过调用 `clmext_destroy_peer` 来删除对等实例。
- 卸载模块之前，主机调用服务函数 `clmext_cleanup`。
- 可在服务器进程关闭时或在不再需要其服务之前卸载模块。

服务器端 API 功能

本节着重介绍服务器端 API 的某些功能：

- 节点类型信息
- 表示不同类型的数据存储的数据类型
- 服务器端共享库
- 文件空间和临时文件
- 用于执行数据库中 SQL 指令的 SQL 回送
- IBM® SPSS® Modeler 与扩展之间的数据模型信息的交换
- 输出文档
- C++ 帮助程序

节点类型

在规范文件中，节点定义采用下列格式：

```
<Node id="identifier" type="node_type" .../>
```

`id` 属性是唯一标识节点的字符串。

`type` 属性将节点标识为下列类型之一：

- 数据读取器

- 数据记录器
- 数据变换器
- 模型构建器
- 模型填充器
- 文档构建器

有关详细信息，请参阅第 9 页码第 2 章中的节点概述。

`clmext_create_peer` 函数将 `Node` 元素的 `id` 和 `type` 两个属性的值作为参数。

单个扩展模型可以实现各种类型的节点，从而执行每种类型节点中的各个函数。例如，一个模块可以实现：

- 数据源的数据读取器和数据记录器
- 各种建模算法的模型构建器和模型填充器
- 各种图形类型的文档构建器

数据和存储类型

对等实例通过在开始执行时在为其提供的迭代器上调用 `clmext_iterator_getRecordValue` 来获取输入数据，并提供输出数据以响应来自主机的 `clmext_peer_getRecordValue` 请求。数据以二进制格式在内存中传送，并且对等和主机必须在该数据类型上达成一致。

二进制数据类型由数据模型决定，并与字段的存储属性有关。

下表列出了可能的存储类型以及用于表示它们的数据类型：

表 9-2
存储类型

存储类型	表示方法	附注
string	char *	字符串始终是 UTF-8 编码
real	CLEMEXTReal	双精度浮点数
integer	CLEMEXTInteger	64 位有符号整数
日期	CLEMEXTDate	64 位有符号整数，表示自 1970 年 1 月 1 日起的天数
时间	CLEMEXTTime	64 位有符号整数，表示自午夜起的秒数
时间戳	CLEMEXTTimestamp	64 位有符号整数，表示自 1970 年 1 月 1 日午夜起的秒数
unknown	-	表示未知数据类型—无法表示值

库

可以在规范文件中声明服务器端共享库以支持节点执行。共享库路径用于定位动态加载到主机进程中的共享库。共享库必须定义所有必需的 API 函数。[有关详细信息，请参阅第 39 页码第 4 章中的共享库。](#)

如果在规范文件中提供了模块名称（在节点定义的 `Execution` 部分），该名称将传递到服务函数 `clmext_create_peer` 的 `nodeId` 参数以创建对等对象。这样，扩展即可创建适当种类的对等模块。`nodeType` 参数值也可能影响创建的对等的种类。模块名称可以为空白，因为一个共享库不会实现一个以上的同一类型的模块。

实现扩展模块的共享库可能会需要从属库。这些从属库应位于与扩展共享库相同的目录中。

临时文件

客户端规范文件和服务器扩展模块可以指定与**文件空间**相关的路径名，文件空间是临时专用空间，对等可以在文件空间中创建执行时使用的文件。文件空间是为对等创建的服务器临时目录的子目录。将根据需要来创建它，并在对等销毁时删除它。

存在文件空间时，对等将完全控制文件空间。文件空间的完整路径名包含在**节点信息文档**中。这是 XML 格式的信息，作为执行 `clemt_node_getNodeInformation` 回调函数的结果返回。[有关详细信息，请参阅第 210 页码节点信息文档。](#)

SQL Pushback

在 SQL 回送中，IBM® SPSS® Modeler 流从 SQL 数据库读取数据并执行数据处理，高级用户可以通过回送 SQL 指令以在数据库本身中执行来提高此操作的效率。

多个标准 SPSS Modeler 节点均支持 SQL 回送，并且服务器端 API 包含函数调用以使 CLEF 节点也可支持 SQL 回送。

`clemt_peer_getSQLGeneration` 服务函数从对等实例生成 SQL，并且可用于将 SQL 执行回送到数据库。对于数据读取器节点，其自身生成的 SQL 必须完整才能创建对等结果集。对于任何其他类型的节点，生成的 SQL 将最有可能从属于为提供对等输入的上游节点生成的 SQL。对等可以通过在其关联节点句柄上调用 `clemt_node_getSQLGeneration` 回调函数来获取上游 SQL。

数据模型处理

某些服务器端 API 调用与 IBM® SPSS® Modeler 和扩展模块之间的数据模型信息交换有关：

- `clemt_node_getDataModel` 获取节点的输入数据模型
- `clemt_peer_getDataModel` 获取对等实例的输出数据模型
- `clemt_node_getOutputDataModel` 获取节点的输出数据模型

其他调用与将数据传入和传出模块的方法有关。数据模型确定用于在下列函数中查找字段值的索引，这些函数返回上一次提取的输入记录中指定字段的值：

- `clemt_peer_getRecordValue`
- `clemt_iterator_getRecordValue`

SPSS Modeler 调用 `clemt_node_getDataModel` 来获取有关输入数据模型的字段信息。信息以 XML 格式返回—例如：

```
<DataModel>
  <Fields>
    <Field name=" abc" storage=" string" type=" set" />
    <Field name=" uvw" storage=" integer" type=" range" />
    <Field name=" xyz" storage=" real" type=" range" />
  </Fields>
```

```
</DataModel>
```

模块可以使用此信息在使用 `clmext_iterator_getRecordValue` 函数检索输入记录的值时提供字段索引，例如：

图片 9-5
输入数据模型示例

索引	0	1	2
字段	abc	uvw	xyz

模块影响输入数据模型的方式受规范文件中 `OutputDataModel` 元素的 `mode` 属性的值控制。模块可以：

- 通过向模型添加新字段来扩展模型。
- 通过删除或重命名现有字段来修改模型。
- 使用新字段替换现有模型。
- 保留模型不变。

下列示例演示扩展和替换模型。

示例一扩展输入数据模型

这是最简单的情况：使模块能够添加新字段并设置其值，但不能删除或更改现有字段的值。

假设规范文件在节点定义中包含下列指令：

```
<OutputDataModel mode="extend">
  <AddField name=" field1" storage=" string" .../>
  <AddField name=" field2" storage=" real" .../>
  ...
</OutputDataModel>
```

此处，将输出数据模型定义为包含输入数据模型中的所有字段以及 `OutputDataModel` 元素中指定的两个附加字段。因此，输出数据模型由五个字段组成。

图片 9-6
输出数据模型扩展输入数据模型的示例

索引	0	1	2	3	4
字段	abc	uvw	xyz	字段 1	字段 2

`clmext_peer_getDataModel` 函数返回仅与添加字段相关的信息，例如：

```
<DataModel>
  <Fields>
  <Field name=" field1" storage=" string" .../>
  <Field name=" field2" storage=" real" .../>
  </Fields>
```



```
</DataModel>
```

返回的信息必须与规范文件中 `<AddField>` 元素的类型和数值（不是名字）相匹配。

模块可以使用回调函数 `clmext_node_getOutputDataModel` 获取 IBM® SPSS® Modeler 期望添加的字的详细信息。此信息可以直接传递回 SPSS Modeler 以响应 `clmext_peer_getDataModel` 的调用。如果用于创建和命名输出字段的规范文件的逻辑比较复杂，这样做会很有用。

当 SPSS Modeler 调用 `clmext_peer_getRecordValue` 时，该模块为每个输出记录提供新值。新字段的字段指数从输入字段的最后一个指数开始。在本示例中，输入数据模型包含三个字段（位于指数位置 0、1 和 2），因此两个输出字段的字段指数为 3 和 4。SPSS Modeler 在调用 `clmext_peer_getRecordValue` 时，不会使字段指数对应于输入字段，因为该模块无法更改这些字段。

示例—替换输入数据模型 (1)

在本示例中，扩展模块将从其输出中丢弃所有输入数据模型字段，并使用新字段替换它们。

规范文件包含下列内容：

```
<OutputDataModel mode="modify">
  <AddField name=" key" storage=" integer" .../>
    <AddField name=" field1" storage=" real" .../>
    <AddField name=" field2" storage=" real" .../>
  ...
</OutputDataModel>
```

这次，通过调用 `clmext_peer_getDataModel` 返回的 XML 数据描述输出数据模型中的所有字段：

```
<DataModel>
  <Fields>
    <Field name=" key" storage=" integer" .../>
    <Field name=" field1" storage=" real" .../>
    <Field name=" field2" storage=" real" .../>
  <Fields>
</DataModel>
```

输出数据模型具有下列结构：

图片 9-7
输出数据模型替换输入数据模型的示例

索引	0	1	2
字段	关键字	字段 1	字段 2

调用 `clmext_peer_getRecordValue` 时使用的字段指数从第一个输出字段 (`key`) 的字段指数 0 开始，然后是下一个字段 (`field1`) 的字段指数 1，依此类推。

示例一替换输入数据模型 (2)

在本示例中，扩展提供的输出数据模型也会替换输入数据模型，如上一示例所示。但是，这次未在规范文件中定义输出数据模型，而是在运行时由服务器上的扩展模块计算输出模型。规范文件包含下列内容：

```
<OutputDataModel mode="modify" method="sharedLibrary" libraryId="myLibraryId" />
```

要计算输出数据模型，IBM® SPSS® Modeler 首先调用 `clemt_peer_configure`，然后调用 `clemt_peer_getDataModel`。如上一示例所示，由 `clemt_peer_getDataModel` 的响应完全定义的输出数据模型中不会自动包含输入数据模型中的任何字段。

注意：在类似扩展模块定义服务器上的输出数据模型的情况下，该模块无法使用 `clemt_node_getOutputDataModel` 获取输出数据模型，因为这样会导致“无效操作”错误。

XML 输出文档

某些服务和回调函数在主机和扩展模块之间以 XML 输出文档的形式传送信息。可以使用多种不同的文档：

XML 输出文档	附注	由对...的调用返回
目录文档	包含与目录相关联的控件值列表。	<code>clemt_peer_getCatalogInformation</code>
数据模型文档	描述进/出节点的字段集	<code>clemt_peer_getDataModel</code> <code>clemt_node_getDataModel</code> <code>clemt_node_getOutputDataModel</code>
错误详细信息文档	提供有关错误或其他状态的信息	<code>clemt_peer_getErrorDetail</code>
执行要求文档	描述对等实例所需的执行支持，如数据缓存或必填输入字段	<code>clemt_peer_getExecutionRequirements</code>
主机信息文档	提供有关主机环境的信息，包含：产品标识符、描述、版本、提供程序、版权和平台详细信息	<code>clemt_host_getHostInformation</code>
模块信息文档	提供有关扩展模块的信息，包含：模块标识符、描述、版本、提供程序、版权和许可详细信息	<code>clemt_getModuleInformation</code>
节点信息文档	提供有关与对等实例相关联的节点的信息，包含：节点标识符、类型和文件空间详细信息	<code>clemt_node_getNodeInformation</code>
参数文档	包含客户端节点的配置参数；内容由扩展确定	<code>clemt_node_getParameters</code>
SQL 生成文档	描述如何将对等的执行转换为 SQL	<code>clemt_peer_getSQLGeneration</code> <code>clemt_node_getSQLGeneration</code>
状态详细信息文档	提供有关进度以及执行期间出现的警告或其他状态的补充信息	<code>clemt_progress_report</code>

目录文档

目录文档描述目录内容，其中包含可以从 UI 控件中显示的值列表。

CLEF 模块执行调用 `getCatalogInformation` 如下：

```
CLEMEXTStatus
getCatalogInformation(
    const char *catalogId,
    char* buffer,
    size_t buffer_size,
    size_t* data_size,
    CLEMEXTErrorCode* errorCode) {
    ...
}
```

其中 `catalogId` 是特定目录的标识，如规范文件的 `Catalog` 元素中的定义。

此函数返回目录文档。

示例

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<CatalogInformation>
  <CatalogEntry>
    <CatalogValue>apples</CatalogValue>
    <CatalogValue>0</CatalogValue>
  </CatalogEntry>
  <CatalogEntry>
    <CatalogValue>oranges</CatalogValue>
    <CatalogValue>1</CatalogValue>
  </CatalogEntry>
  <CatalogEntry>
    <CatalogValue>bananas</CatalogValue>
    <CatalogValue>2</CatalogValue>
  </CatalogEntry>
</CatalogInformation>
```

数据模型文档

数据模型文档描述进/出节点的数据模型，即包含字段名称、类型和相关信息的字段集。它包含类型节点的可用信息。

不消耗输入的对等（源节点）具有空输入模型，而不生成输出的对等（终端节点）具有空输出模型。消耗输入并生成输出的对等（过程节点）必须知道如何从其输入计算其输出模型。

对等可以通过在其关联节点句柄上调用 `cleMext_node_getDataModel` 来获取其输入数据模型。对等提供其输出数据模型以响应来自主机的 `cleMext_peer_getDataModel` 请求。

任何数据模型均可以直接表示为数据字典，该数据字典枚举数据模型中的所有字段及其属性。节点向对等提供的输入数据模型始终都采用这种形式。对等生成的输出数据模型可能会具有相同的形式，也可能表示为应用于输入模型的一系列操作（添加字段、删除字段、修改字段）。这将大大简化某些节点的输出模型。

数据模型文档中的字段顺序非常重要，该顺序决定了数据在相应的输入和输出数据集中的出现顺序。

数据模型可能不完整，并且仅提供数据的部分规范。完整指定的以允许对等计算执行计划的输入模型对于该对等来说是**可执行的**。可执行的数据模型必须包含每一个字段的二进制类型，以便能够正确整理输入和输出数据。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<DataModel>
  <Fields>
    <Field name="Age" type="range" storage="integer" direction="in">
      <Range minValue="15" maxValue="74"/>
    </Field>
    <Field name="Sex" type="flag" storage="string">
      <Values>
        <Value value="F" flagValue="false" displayLabel="Female"/>
        <Value value="M" flagValue="true" displayLabel="Male"/>
      </Values>
    </Field>
    <Field name="BP" type="orderedSet" storage="integer">
      <Values>
        <Value value="-1" />
        <Value value="0" />
        <Value value="1" />
      </Values>
    </Field>
    <Field name="Cholesterol" type="flag" storage="string">
      <Values>
        <Value value="NORMAL" flagValue="false"/>
        <Value value="HIGH" flagValue="true"/>
      </Values>
    </Field>
    <Field name="Na" type="range" storage="real" displayLabel="Blood sodium">
      <Range minValue="0.500517" maxValue="0.899774"/>
    </Field>
    <Field name="K" type="range" storage="real" displayLabel="Potassium concentration">
      <Range minValue="0.020152" maxValue="0.079925"/>
    </Field>
    <Field name="Drug" type="set" storage="string" direction="out">
      <Values>
        <Value value="drugA"/>
        <Value value="drugB"/>
        <Value value="drugC"/>
        <Value value="drugX"/>
        <Value value="drugY"/>
      </Values>
    </Field>
  </Fields>
</DataModel>
```

```

    </Field>
  </Fields>
</DataModel>

```

错误详细信息文档

错误详细信息文档用于将消息（错误、警告、信息）发送回 IBM® SPSS® Modeler，并提供有关错误或其他状态的信息。扩展模块可以提供错误详细信息文档以解释模块特定错误，从而响应来自主机的 `clemt_peer_getErrorDetail` 请求。

错误详细信息是一个或多个 `Diagnostic` 元素的集合，其中每一个诊断均至少包含一个错误代码、一条消息，以及一个或多个包含要在消息中插入的进一步信息的参数集合。错误代码与规范文件中 `StatusCode` 元素的值相匹配。

消息可能会具有不同的语言变量，或者客户端可以使用错误代码从资源包中选择本地化后的消息。一系列诊断元素描述导致错误的因果链。

示例

```

<?xml version="1.0" encoding="utf-8"?>
<ErrorDetail>
  <Diagnostic code="123" severity="error">
    <Message>You can't do that ({0})</Message>
    <Parameter>Permission denied</Parameter>
  </Diagnostic>
  <Diagnostic code="456" severity="warning">
    <Message>That was silly!</Message>
    <Message lang="fr">Quel idiot!</Message>
  </Diagnostic>
</ErrorDetail>

```

执行要求文档

执行要求文档描述对等实例所需的执行支持。对等实例可以提供执行要求文档以响应来自主机的 `clemt_peer_getExecutionRequirements` 请求。主机在对等上调用 `clemt_peer_beginExecution` 之前查询要求文档，以便提供正确的执行环境。

主机可以提供数据缓存服务，以使模块能够使用 `clemt_iterator_rewind` 函数跨输入数据进行多次传递。

示例

```

<?xml version="1.0" encoding="utf-8"?>
<ExecutionRequirements>
  <Cache/><!-- this ensures that the CLEF module can make multiple passes over the input data -->
</ExecutionRequirements>

```

主机信息文档

主机信息文档描述主机环境。扩展模块可以通过在主机句柄上调用 `clemt_host_getHostInformation` 来获取主机信息。

返回的信息包含产品标识符、描述、版本、提供程序、版权和平台的详细信息。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<HostInformation>
  <Host name="clemlocal" externalEncoding="cp1252" language="english_us"
    locale="English_United Kingdom.1252" provider="IBM Corp." version="15" platform="Windows XP SP2"
    copyright="Copyright 1995-2011 IBM Corp. All rights reserved.">
<VersionDetail major="12" minor="0"/>
  <PlatformDetail osType="windows" osName="WindowsNT" osMajor="5" osMinor="1"/>
  <LibraryDetail path="C:\Program Files\IBM\SPSS\Modeler\15\ext\bin\my.module\myModule.dll"/>
</Host>
</HostInformation>
```

模块信息文档

模块信息文档描述扩展模块。扩展模块必须提供模块信息文档以响应来自主机的 `clemtxt_getModuleInformation` 请求。

返回的信息包含模块标识符、描述、版本、提供程序、版权和许可的详细信息。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<ModuleInformation>
  <Module name="MyModule" provider="My Company Inc." version="10.1.0.329"
    copyright="Copyright 2006 My Company Inc. All rights reserved.">
<VersionDetail major="10" minor="1" release="0" build="329"/>
  <Licence code="1234" type="mandatory"/>
  <Description>Provides a thorough test of the new extensions framework.</Description>
</Module>
</ModuleInformation>
```

节点信息文档

节点信息文档描述与对等实例相关联的节点。对等实例可以通过在节点句柄上调用 `clemtxt_node_getNodeInformation` 来获取节点信息。节点信息包含节点标识符、类型和文件空间的详细信息。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<NodeInformation>
  <Node name="databaseImport" type="dataReader">
    <FileSpace path="C:\Program Files\IBM SPSS Modeler Server15\tmp\ext-8005-6711-01"/>
  </Node>
</NodeInformation>
```

参数文档

参数文档包含规范文件中定义的每一个 **Property** 元素的详细信息。这些详细信息以配置参数的形式返回，对等可以通过在节点句柄上调用 `clmext_node_getParameters` 来获取这些配置参数。

参数具有名称和值，其中的值可以为：

- 简单值（字符串）
- 键控值（键和值）
- 结构化值（指名值的集合）
- 值列表

参数文档的内容完全由扩展数据包决定。参数定义在客户端规范文件中，并由服务器扩展模块解释。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
  <Parameter name="linesToScan" value="50"/>
  <Parameter name="useCaption" value="true"/>
  <Parameter name="caption" value="My Caption"/>
  <Parameter name="captionPosition" value="north"/>
  <Parameter name="defaultAggregation">
    <ListValue>
      <Value value="min"/>
      <Value value="max"/>
      <Value value="mean"/>
      <Value value="stddev"/>
    </ListValue>
  </Parameter>
</Parameters>
```

SQL 生成文档

SQL 生成文档描述如何将对等的执行转换为 SQL。

对等可以提供 SQL 生成文档，以响应来自主机的 `clmext_peer_getSQLGeneration` 请求。主机会尝试先执行 SQL 然后再在内部执行对等。

消耗输入的对等可以通过在其关联节点句柄上调用 `clmext_node_getSQLGeneration` 来获取其输入 SQL。

SQL 生成文档的主要组成部分是复制节点或流段的执行行为的 SQL 语句。对于生成数据的节点（即数据读取器节点或数据转换器节点），该语句必须是 **SELECT** 语句，并且必须与将数据模型中的字段名称映射到 **SELECT** 语句中的列名称的字典同时使用。

SQL 生成文档还可以包含数据库连接的属性（如数据源名称和产品名），将凭借该数据库连接执行该语句。对等可以使用这些属性帮助确定其生成的 SQL。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<SqlGeneration>
  <Properties
    datasourceName="SQL Server"
    databaseName="DataMining"
    serverName="GB1-RDUNCAN1"
    passwordKey="PW0"
    userName="fred"
    dbmsName="Microsoft SQL Server"
    dbmsVersion="09.00.1399"/>
  <Statement>
    <Bindings>
      <Binding columnName="C0" fieldName="ID"/>
      <Binding columnName="C1" fieldName="START_DATE"/>
    </Bindings>
    <TableParameters>
      <TableParameter name="{TABLE26}" value="dbo.DRUG4N"/>
    </TableParameters>
    <Sql>
      SELECT
      T0.ID AS C0, T0."START_DATE" AS C1
      FROM ${TABLE26} T0
      WHERE (T0."START_DATE" > '2003-01-01')
      ORDER BY 2 ASC
    </Sql>
  </Statement>
</SqlGeneration>
```

状态详细信息文档

状态详细信息文档提供有关进度以及执行期间出现的非致命警告或其他状态的信息。扩展模块可以使用 `clemext_progress_report` 回调函数异步调度状态详细信息文档。

状态详细信息文档由一个或多个 **Diagnostic** 元素的集合组成，其中每一个诊断至少包含一个状态代码、一条消息（除非属性文件中提供），以及一个或多个包含要在消息中插入的进一步信息的参数集合。**StatusDetail** 元素也可以有可选 **destination** 属性，将消息传递到下列位置之一的目标：

- 由 IBM® SPSS® Modeler 管理的本地跟踪文件
- 客户端（适用于与用户有关的消息）
- 所有（发动到所有可能的目标）

Diagnostic 元素的格式为：

```
<Diagnostic code="integer" severity="severity_level">
  <消息>message_text</Message>
  <Parameter>value</Parameter>
</Diagnostic>
```

其中：

`code`（必填）是一个表示状态代码的整数。

`severity` 表示条件的严重性，是下列内容之一：`unknown`、`information`、`warning`、`error` 或 `fatal`。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<StatusDetail destination="client">
  <Diagnostic code="654" severity="information">
    <Message>Processed {0} records</Message>
    <Parameter>10000</Parameter>
  </Diagnostic>
</StatusDetail>
```

使用本地化消息

如果您想从属性文件中使用本地化消息，您从状态详细信息文档中忽略 `Message` 元素，并使用规范文件中的消息密钥，如以下示例所示：

```
...
<Execution ...>
...
  <StatusCodes>
    ...
    <StatusCode code="21" status="warning" messageKey="fieldIgnoredMsg.LABEL"/>
    ...
  </StatusCodes>
</Execution>
...
```

`messages.properties` 文件将会包含：

```
fieldIgnoredMsg.LABEL=Field "{0}" cannot be used for model building and was ignored
```

在状态详细信息文档中，您然后将要替换的参数（如字段名）发送到本地化消息中，例如：

```
<?xml version="1.0" encoding="utf-8"?>
<StatusDetail>
  <Diagnostic code="21">
    <Parameter>BP</Parameter>
  </Diagnostic>
</StatusDetail>
```

C++ 帮助程序

某些 CLEF 示例节点包含许多预定义的 C++ 源文件，称为**辅助程序**。这些辅助程序相当于某些基于 C 的服务器端 API 的包装，可以轻松地将它们编译到 C++ CLEF 中。

表 9-3
C++ 帮助程序

帮助程序	描述
BufferHelper	管理 C API 中使用的可调整大小的内存缓冲区
DataHelper	帮助包装数据读取和写入操作
HostHelper	包装 CLEMEXT HostHandle 对象
XMLHelper	包装 XML 处理

这些辅助程序采用 .cpp 和 .h 文件对的形式—例如 BufferHelper.cpp 和 BufferHelper.h。

有关查看这些辅助程序文件的信息，请参阅[检查源代码第 33 页码](#)。

有关这些文件的更详细说明可以在服务器端 API 文档中找到，如下所示：

- ▶ 在“CLEF API 文档”屏幕中，选择服务器端 API 概述。
- ▶ 单击文件选项卡。
- ▶ 单击与您要查看其信息的辅助程序相对应的 .h 文件名。
- ▶ 在数据结构下，单击对应的类名即可显示该文档。

有关访问 CLEF API 文档的信息，请参阅[CLEF API 文档第 193 页码](#)。

错误处理

每一个 API 函数调用均返回一个状态代码 (CLEMEXTStatus) 和一个可选的模块特定错误代码 (CLEMEXTErrorCode)。状态代码可以是 success (无错误) 或 API 函数枚举的错误代码之一；这些错误代码几乎总是包含“模块特定错误”。模块特定错误代码可以为 0，表示“无模块特定错误”。

状态代码消息由 IBM® SPSS® Modeler 提供。通用状态代码的详细信息可在服务器端 API 文档中找到，如下所示：

- ▶ 在“CLEF API 文档”屏幕中，选择服务器端 API 概述。
- ▶ 单击模块选项卡。
- ▶ 选择通用状态代码。

有关访问 CLEF API 文档的信息，请参阅[CLEF API 文档第 193 页码](#)。

模块特定错误消息可以通过下列方式提供：

- 在规范文件中（在模块部分的 StatusCodes 元素中）
- 在规范文件中引用的资源包中
- 通过扩展模块

对于模块特定错误代码，模块可以提供额外的错误详细信息，其中包含默认错误消息（适用于不属于规范文件或资源包的错误）和要插入到消息中的参数。多个错误消息可以描述导致错误的因果链。

客户端上的错误报告采用下列格式：

```
node_label:message
```

其中：

- node_label 是在其中指定该模块的 **Node** 元素的 **label** 属性值。
- message 是消息的文本，既可以由服务器提供，也可以在规范文件（或用于本地化的 .properties 文件）中定义。

XML 解析 API

IBM® SPSS® Modeler 包含 Apache 的 Xerces-C XML 解析程序，该解析程序提供了许多允许模块读取和写入 XML 数据的回调。如果需要，可以替代自己的 XML 解析程序。

使用服务器端 API

要在节点中包含服务器端函数调用：

- ▶ 创建包含函数调用的 C++ .cpp 和 .h 源文件。
- ▶ 将源文件编译到动态链接库（.dll）文件中。
- ▶ 在规范文件中包含对 .dll 文件的引用—例如：

```
<Resources>
.
  <SharedLibrary id="mylib1" path="mycorp.mynode/mylib" />
.
</Resources>
```

有关详细信息，请参阅第 39 页码第 4 章中的共享库。

您还会发现看一看与此版本一同提供的示例节点的源代码也会很有用。有关详细信息，请参阅第 33 页码第 3 章中的检查源代码。

服务器端编程准则

CLEF 模块的服务器端动态链接库（DLL）部分应该遵守许多准则，以确保模块功能正确并避免影响 IBM® SPSS® Modeler 的操作。CLEF 模块应该：

- 确保对等执行自包含
- 支持单个过程中的多个对等实例
- 确保线程安全
- 避免修改线程或过程环境
- 限制模块内的线程用途

- 正确处理请求以取消执行
- 重启中断的系统调用 (UNIX)
- 调用 `CoInitialize` 或 `CoUninitialize` 时务必注意 (Windows)
- 避免假设卸载模块的时间
- 启动子过程时务必注意
- 避免写入标准输出或标准误

以下部分更加详细介绍了这些区域中的每一个。

确保对等执行自包含

对等实例不得假设在 IBM® SPSS® Modeler 服务器过程内存在其他对等实例。SPSS Modeler 可能计划执行，以便在流中直接相邻的与节点对应的对等实例实际上按不同阶段执行，以便实例的存在和执行不会重叠。

对等实例因此应该自包含，不得尝试直接与其他对等实例通信，例如通过管道或套接字。不同对等实例之间的所有通信应该通过读取或将数据写入流中直接执行，或通过一些外部代理（例如管理对等之间共享数据的数据库服务器）间接执行。

支持单个过程中的多个对等实例

最终用户可以当执行流时在服务器过程中创建特定 CLEF 模块的多个对等实例（例如相同类型的多个节点）。因此 CLEF 模块中的任何静态数据在多个对等实例中共享，不得用于存储对等对象的专有数据。静态数据的示例是 C++ 类的静态成员，以及 C 编译单位中的全局或静态变量。

CLEF 模块 API 函数必须重新进入并且避免进行任何非重新进入系统调用。例如，当对等实例使用 `clefext_iterator_nextRecord` 从其输入迭代器中获取输入数据时，这可能在位于第一个对等上游并且生成第一个对等最终使用的数据的第二个对等实例上依次调用 `clefext_peer_nextRecord`。

`strtok` 等系统调用不是重新进入，不得使用。有关重新进入选项的详细信息，请参考操作系统文档了解您使用的平台。

确保线程安全

IBM® SPSS® Modeler 可能从不同执行线程中插入执行多个对等实例。因此应该保护访问任何对等对象之间共享的资源，例如通过与互斥体（互相排斥对象）同步或类似线程库服务。

CLEF 模块必须避免进行任何线程并不安全的系统调用。有关更多信息，请参考您的操作系统文档或 UNIX `man` 页面。

避免修改线程或过程环境

避免使用可能更改调用线程或过程环境的系统调用。

这种调用的一些示例在此列出 - 列表绝不会是穷举：

- `setlocale` 当用于更改区域而非读取区域信息

- `SetCurrentDirectory` (Windows) 或 `chdir` (UNIX)
- `LogonUser` (Windows) 或 `seteuid` (UNIX)
- `putenv`
- `exit`
- `signal`

注意：在 Windows 上，一个更改线程环境但可能必要的调用是 `CoInitialize`。有关详细信息，请参阅第 218 页码调用 `CoInitialize` 或 `CoUninitialize` 时务必注意 (Windows)。

限制模块内的线程用途

通常，模块可自由内部使用线程。然而，应该只在 IBM® SPSS® Modeler 用于调用 CLEF 模块函数的线程上调用回调函数（除 `clemext_peer_cancelExecution` 外）。

可以从在模块内执行的任何线程中异步调用以下回调函数：

- `clemext_progress_report`
- `clemext_channel_send`

对等实例应该确保多个线程不会同时调用每个调用。

正确处理请求以取消执行

当最终用户请求取消执行对等实例时，IBM® SPSS® Modeler 对模块的 `clemext_peer_cancelExecution` 函数进行异步调用。开发人员应该尝试执行此调用。注意此函数将被异步调用，并在正由模型执行另一个 CLEF API 函数调用时调用。

重启中断的系统调用 (UNIX)

在 UNIX 上，IBM® SPSS® Modeler 应用程序使用信号和信号处理程序。如果过程在执行调用期间收到一个信号，一些 UNIX 系统调用可能返回代码 `EINTR`—查看 `man` 页面了解特定 UNIX 平台上的系统调用。

如果发生此活动，调用代码应该查看 `EINTR` 返回代码并重启调用。达到这点的一种方法是创建一个简单换行函数 (`open_safe`) 并且使换行的应用程序调用：

```
int
open_safe(const char* path, int oflag, mode_t mode) {
    int res;
    while ((res = ::open(path, oflag, mode)) == -1
           && errno == EINTR) {
    }
    return res;
}
```

调用 `CoInitialize` 或 `CoUninitialize` 时务必注意 (Windows)

在 Windows 上, 需要使用 Windows 组件对象模型 (COM) 库服务的线程应该调用系统 API 函数 `CoInitialize` 然后使用 COM 服务, 并且应该在完成时调用 `CoUninitialize`。IBM® SPSS® Modeler 为一个模块调用 CLEF API 的线程可能或不可能已经拥有调用的 `CoInitialize`。

希望从这些线程中使用 COM 服务的 CLEF 模块应该调用 `CoInitialize`, 通常在 `clemt_create_peer` 或 `clemt_peer_beginExecution` 函数中。如果该调用成功, 模块必须稍后当线程完成执行时还调用 `CoUninitialize`, 通常分别在 `clemt_destroy_peer` 或 `clemt_peer_endExecution` 中。

有关 `CoInitialize` 调用的更多信息, 请参阅 Microsoft Developer Network (MSDN) 站点 <http://msdn.microsoft.com> 上的文档。

避免假设卸载模块的时间

当前, 始终加载 CLEF 模块, 直到会话结束 (例如模块不能按需要卸载和重新载入)。函数 `clemt_cleanup` 始终未被调用, 即使从加载模块的 IBM® SPSS® Modeler 服务器过程中退出。因此开发人员任何时候都不得假定模块将被卸载, 其资源已释放。

启动子过程时务必注意

通过 `CreateProcess` (Windows) 或 `fork` (UNIX) 启动子过程可以引入大量复杂情况, 使父过程和子过程进行交互, 使子过程继承父过程中开放的资源。

如果 CLEF 模块需要调用超出过程的执行, 请考虑使用一个适当的其他体系结构。例如, CLEF 模块可能使用应用程序服务器提供的服务以执行所需任务。

具体来说, Windows 过程应该避免使用 `CreateProcess` 函数启动子过程, 将 `blinheritHandles` 参数设为 `TRUE`。这样做使子过程继承在父 (IBM® SPSS® Modeler 服务器) 过程中开放的所有文件描述符。

避免写入标准输出或标准误

如果 CLEF 模块写入一个过程的标准输出或标准误流 (可能出于调试目的), 通常最终用户不会看到。然而, 当使用 IBM® SPSS® Modeler Solution Publisher 部署并从命令行命令解释程序中执行包含 CLEF 节点的流时 (无论是在 Windows 还是 UNIX 中), 此输出将可见, 可能使用户感到混乱。

CLEF 模块可以通过调用主机回调函数 `clemt_host_trace` 并传递消息以字符串的形式显示来调用跟踪服务。还需要通过在 IBM® SPSS® Modeler 安装目录中的 IBM® SPSS® Modeler Server 配置选项文件 (`/config/options.cfg` 中使用以下设置在 SPSS Modeler 安装中启用跟踪:

```
trace_extension, 1
```

跟踪的消息输出到 SPSS Modeler 安装目录中的 `/log/trace-<process_ID>-<process_ID>.log` 文件, 其中 `process_ID` 是 SPSS Modeler Server 进程的标识符。避免同时跟踪多个会话, 因为他们都共享相同的日志文件。

测试和分发

测试 CLEF 扩展

强烈建议先在本地测试新扩展，然后再将其分发给其他用户。

创建完规格文件和任何关联资源包、.jar 文件、共享库和用户帮助文件后，您可以将文件排列为所需文件结构并复制它们到您的本地 IBM® SPSS® Modeler 安装中以测试扩展。下次启动 SPSS Modeler 时，您会在 SPSS Modeler 用户界面中看到新扩展。

测试 CLEF 扩展

- ▶ 如果 IBM® SPSS® Modeler 为打开的话，请关闭它。
- ▶ 如果扩展定义 CLEF 节点或输出，我们建议激活扩展对话框的“调试”选项卡，直到扩展正确工作。 [有关详细信息，请参阅第 220 页码使用“调试”选项卡。](#)
- ▶ 将客户端和服务端文件排列为所需结构。确保节点所需的所有关联资源（例如，.jar 或 .dll 文件）都已复制到正确的位置。 [有关详细信息，请参阅第 5 页码第 1 章中的文件结构。](#)
- ▶ 复制客户端目录到 SPSS Modeler 安装目录的 \ext\lib 文件夹。
- ▶ 复制服务端目录到 SPSS Modeler 安装目录的 \ext\bin 文件夹。
- ▶ 开始 SPSS Modeler。
- ▶ 如果该扩展定义了菜单或菜单项，请确保这些内容可以在主菜单系统中正确显示。如果扩展定义了新节点，确保节点显示在规范文件中的正确节点选项板上的所需位置。
- ▶ 充分测试扩展。

例如，确保：

- 节点性能不随字段数量和记录数量的增加而下降
 - 对空值采用一致的处理方式
 - 如果需要，可支持不同区域（例如，欧洲和远东）
- ▶ 一旦添加了扩展，您还可以对其规格文件进行更改。然而，您所进行的任何更改只在您重启 SPSS Modeler 后生效。

调试 CLEF 扩展

CLEF 提供了以下辅助调试扩展的功能：

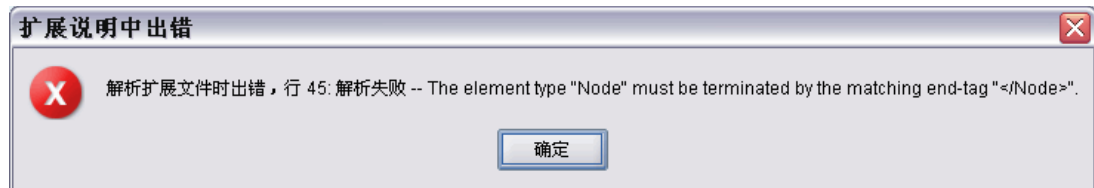
- XML 语法错误消息

- 外部执行
- 调试选项卡

XML 语法错误

XML 解析程序中的错误消息标记了规范文件中的错误 XML 语法，例如：

图片 10-1
XML 语法错误



此消息显示错误所在的大概行号以及错误的类型。

要从这种情况下恢复：

- ▶ 更正文件中的错误。
- ▶ 按照[测试 CLEF 扩展第 219 页码](#)中的过程重新测试文件。
- ▶ 重复此过程，直到规范文件中没有语法错误。

外部执行

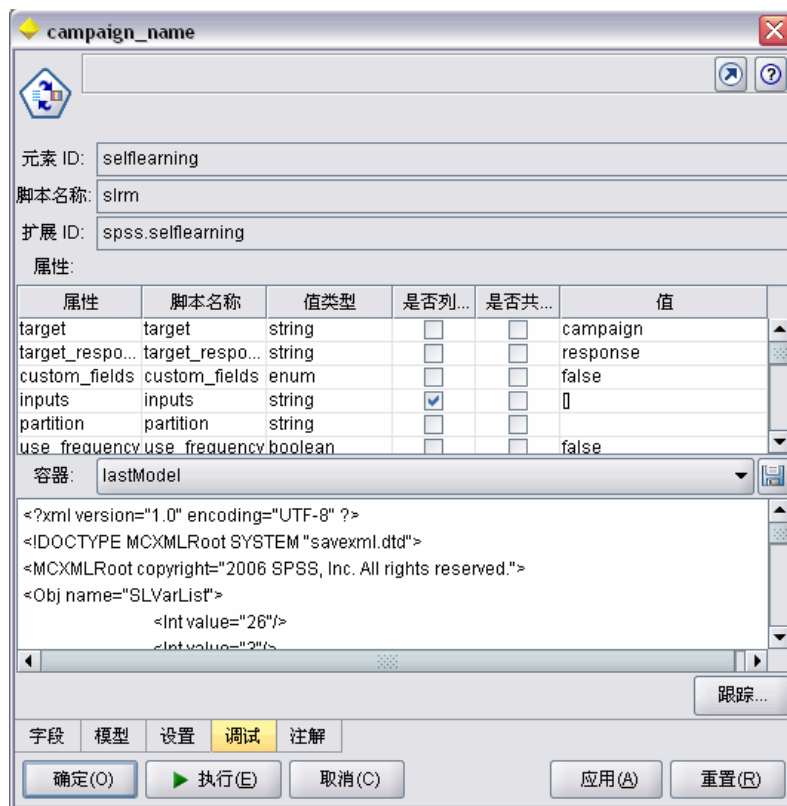
通常，用户所编写的 CLEF 扩展均是在其自身的过程中执行，而不是在 IBM® SPSS® Modeler 过程执行。这样做有助于调试，即使某个外部过程失败，也不会破坏掉整个 IBM® SPSS® Modeler Server 过程。

注意：可以覆盖此默认设置。[有关详细信息，请参阅第 222 页码更改执行选项。](#)

使用“调试”选项卡

对于所有与 CLEF 节点或输出关联的对话框或框，都可以通过激活“调试”选项卡检查相应对象的属性设置。也可以查看在扩展中定义的所有容器的内容，并可将这些内容保存到文件中，以供进一步检查。[有关详细信息，请参阅第 59 页码第 4 章中的 Containers。](#)

图片 10-2
对话框中的“调试”选项卡



可以通过将规范文件中 `Extension` 元素的 `debug` 属性值设置为 `true` 来激活“调试”选项卡。有关详细信息，请参阅第 36 页码第 4 章中的扩展元素。

选项卡上的字段包括：

元素 ID。 扩展的唯一标识符 – 规范文件中 `ExtensionDetails` 元素的 `id` 属性的值。

脚本名称。 在脚本中引用节点时的唯一标识符 – `Node` 元素 `scriptName` 属性的值。

扩展 ID。 扩展文件夹的名称，该扩展文件夹中包含扩展文件和目录资源。该值由 `ExtensionDetails` 元素的 `providerTag` 和 `id` 属性（由“.”字符分隔）连接而成。请注意，对于标识符的 `providerTag` 部分，该值不应该包含供内部使用的字符串 `spss`。

属性。 此表显示节点的 `Property` 声明的选定详细信息：

- **属性。** 属性的唯一识别符 – `Property` 元素 `name` 字段的值。
- **脚本名称。** 在脚本中引用属性的唯一标识符 – `Property` 元素 `scriptName` 属性的值。
- **值类型。** 此属性可采用的值类型，由 `Property` 元素的 `valueType` 属性定义。
- **列表？** 表示属性是否为指定值类型的值列表 – `Property` 元素 `isList` 属性的值。

- **共享?** 如果选中，表示扩展中不止一处使用了此属性（例如：模型构建器节点、模型输出和模型填充器等）。
- **值。** 属性的默认值（如果有）。

容器。 显示选定容器的内容（例如：模型数据）。单击该字段可显示为扩展定义的所有其他容器的列表，选择不同容器可显示不同的内容。单击相邻的**保存容器**按钮可以 XML 格式保存选定容器的内容，以供进一步检查。

跟踪。 显示执行节点时可查看跟踪输出的对话框。

更改执行选项

默认情况下，用户编写的 CLEF 扩展模块在独立于主 IBM® SPSS® Modeler 过程的过程中执行。因此，即使扩展过程失败，也不会导致 SPSS Modeler 过程失败。与此不同，IBM Corp. 提供的模块默认在主过程中执行。

通过提供两种服务器配置选项，系统管理员可以将一个或更多已命名的模块从其中一种模式更改为相反的模式。两个选项都采用逗号分隔的模块标识符列表来指示哪些模块受到了更改的影响。

注意：正常情况下，仅在客户支持代表请求时才更改这些选项之一。

选项如下所示：

过程执行选项

此选项允许一般会载入到外部过程的扩展模块（通常为用户编写的模块）可直接载入 SPSS Modeler。格式为：

```
clef_inprocess_execution, "  
moduleID1[  
moduleID2[...  
moduleIDn]]"
```

其中，moduleID 是相关规范文件中 `ExtensionDetails` 元素的 `id` 属性值。示例如下：

```
clef_inprocess_execution, "test.example_filereader"
```

外部执行选项

此选项允许一般被直接载入到 SPSS Modeler 的扩展模块（通常为 IBM Corp. 提供的模块）载入到外部过程。格式为：

```
clef_external_execution, "  
moduleID1[  
moduleID2[...  
moduleIDn]]"
```

其中，moduleID 与 `clef_inprocess_execution` 的相同。（虚构的）示例如下：

```
clef_external_execution, "spss.naivebayes, spss.terminator"
```

添加或更改执行选项

要添加或更改执行选项，请遵循《SPSS Modeler 15 服务器管理与性能指南》上“使用 options.cfg 文件”中所述的步骤。

分发 CLEF 扩展

新的扩展全部测试完毕并可进行分发，过程如下：

- ▶ 如果“调试”选项卡已激活的话，取消激活。 [有关详细信息，请参阅第 220 页码使用“调试”选项卡。](#)
- ▶ 创建一个文件结构，准确说明要安装扩展文件的方式。 [有关详细信息，请参阅第 5 页码第 1 章中的文件结构。](#)
- ▶ 将该文件结构压缩为 .zip 文件。您会发现为客户端和服务端文件单独制作单独 .zip 文件更为简便。
- ▶ 向最终用户分发 .zip 文件。

安装 CLEF 扩展

要安装 CLEF 扩展：

- ▶ 收到包含扩展文件结构的 .zip 文件时，将客户端文件解压到 IBM® SPSS® Modeler 安装目录下的 \ext\lib 文件夹。
- ▶ 将服务器端文件解压到 SPSS Modeler 安装目录（如果使用 IBM® SPSS® Modeler Server，则为相应的安装目录）下的 \ext\bin 文件夹中。
- ▶ 启动 IBM® SPSS® Modeler 并检查在节点选项板上指定的位置是否显示新节点。

卸载 CLEF 扩展

要卸载 CLEF 扩展：

- ▶ 在 IBM® SPSS® Modeler 安装目录下的 \ext\lib 目录中找到扩展文件夹。
如果扩展还安装了服务器端扩展文件夹，可在 SPSS Modeler 或 IBM® SPSS® Modeler Server 安装目录下的 \ext\bin 目录中找到此文件夹。
- ▶ 删除一个或多个扩展文件夹。

下次启动 IBM® SPSS® Modeler 时，更改才生效。

CLEF XML 架构

CLEF 元素参考

本节提供 CLEF 中所有元素的参考。

每个主题都列出了元素及其父元素和子元素的有效属性。图表中显示该元素的所有子元素。请注意，图表中的箭头指示可在其他元素间共享的元素。这些元素作为此主题（“CLEF 元素参考”）的子主题，而不是作为父主题的子主题列在目录中。

Action 元素

表 A-1
属性 Action

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
id	必须的		string
imagePath	optional		string
imagePathKey	optional		string
label	必须的		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string
shortcut	optional		string
shortcutKey	optional		string

XML 表示

```

<xs:element name="Action">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="imagePath" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="imagePathKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="shortcut" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="shortcutKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Actions 元素](#)

ActionButton 元素

表 A-2
属性 ActionButton

属性	使用	说明	有效值
action	必须的		string
showIcon	optional		boolean
showLabel	optional		boolean

XML 表示

```
<xs:element name="ActionButton">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="action" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"/></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional"
    default="true"/></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#), [ModelViewerPanel 元素](#),
[SelectorPanel 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#),
[TextBrowserPanel 元素](#)

Actions 元素

XML 表示

```
<xs:element name="Actions">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Action"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

父 元素

CommonObjects 元素

子 元素

Action 元素

AddField 元素表 A-3
属性 AddField

属性	使用	说明	有效值
direction	可选的		in out both none partition
directionRef	optional		
fieldRef	optional		
group	optional		string
label	可选的		string
missingValuesRef	optional		
name	必须的		
prefix	optional		string
role	optional		unknown predictedValue predictedDisplay- Value probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
storage	可选的		unknown integer real string date time timestamp
storageRef	optional		
tag	optional		string
targetField	optional		string

属性	使用	说明	有效值
type	可选的		auto range discrete set orderedSet flag typeless
typeRef	optional		
value	optional		string

XML 表示

```

<xs:element name="AddField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"></xs:element>
      <xs:element ref="Values" minOccurs="0"></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
          <xs:element ref="Range" minOccurs="0"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION"
        minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="real"></xs:enumeration>
    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="time"></xs:enumeration>
    <xs:enumeration value="timestamp"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"></xs:enumeration>
    <xs:enumeration value="range"></xs:enumeration>
    <xs:enumeration value="discrete"></xs:enumeration>
    <xs:enumeration value="set"></xs:enumeration>
    <xs:enumeration value="orderedSet"></xs:enumeration>
    <xs:enumeration value="flag"></xs:enumeration>
    <xs:enumeration value="typeless"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"></xs:attribute>

```

```

<xs:attribute name="fieldRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
<xs:attribute name="storageRef" type="EVALUATED-STRING"
use="optional"></xs:attribute>
<xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
<xs:attribute name="directionRef" type="EVALUATED-STRING"
use="optional"></xs:attribute>
<xs:attribute name="missingValuesRef" type="EVALUATED-STRING"
use="optional"></xs:attribute>
<xs:attribute name="role" type="MODEL-FIELD-ROLE" use="optional">
  <xs:enumeration value="unknown"></xs:enumeration>
  <xs:enumeration value="predictedValue"></xs:enumeration>
  <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
  <xs:enumeration value="probability"></xs:enumeration>
  <xs:enumeration value="residual"></xs:enumeration>
  <xs:enumeration value="standardError"></xs:enumeration>
  <xs:enumeration value="entityId"></xs:enumeration>
  <xs:enumeration value="entityAffinity"></xs:enumeration>
  <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
  <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
  <xs:enumeration value="propensity"></xs:enumeration>
  <xs:enumeration value="value"></xs:enumeration>
  <xs:enumeration value="supplementary"></xs:enumeration>
</xs:attribute>
<xs:attribute name="targetField" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="value" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="group" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="tag" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="prefix" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[ForEach 元素](#), [ModelFields 元素](#)

子 元素

[MissingValues 元素](#), [ModelField 元素](#), [NumericInfo 元素](#), [Range 元素](#), [Values 元素](#)

相关 元素

[ChangeField 元素](#)

MissingValues 元素

表 A-4
属性 MissingValues

属性	使用	说明	有效值
treatNullAsMissing	optional		boolean
treatWhitespaceAsMissing	optional		boolean

XML 表示

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>

```



```

    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="Range" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
</xs:element>

```

父 元素

[AddField 元素](#)

子 元素

[Range 元素](#), [Values 元素](#)

ModelField 元素

表 A-5
属性 ModelField

属性	使用	说明	有效值
group	可选的		
role	必须的		unknown predictedValue predictedDisplay- Value probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	可选的		string
targetField	可选的		string
value	可选的		string

XML 表示

```

<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="predictedValue"></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
    <xs:enumeration value="probability"></xs:enumeration>
    <xs:enumeration value="residual"></xs:enumeration>
    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

```

    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
  <xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[AddField 元素](#)

And 元素

XML 表示

```

<xs:element name="And">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

父 元素

[Command 元素](#), [Constraint 元素](#), [CreateDocument 元素](#), [CreateDocumentOutput 元素](#), [CreateInteractiveDocumentBuilder 元素](#), [CreateInteractiveModelBuilder 元素](#), [CreateModel 元素](#), [CreateModelApplier 元素](#), [CreateModelOutput 元素](#), [Enabled 元素](#), [Not 元素](#), [Option 元素](#), [Or 元素](#), [Run 元素](#), [Visible 元素](#)

子 元素

[And 元素](#), [Condition 元素](#), [Not 元素](#), [Or 元素](#)

Attribute 元素

表 A-6
属性 Attribute

属性	使用	说明	有效值
defaultValue	optional		
description	optional		string
descriptionKey	optional		string
isList	optional		boolean

属性	使用	说明	有效值
label	必须的		string
labelKey	optional		string
name	必须的		string
valueType	可选的		string encryptedString integer double boolean date enum

XML 表示

```

<xs:element name="Attribute">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="valueType" type="ATTRIBUTE-VALUE-TYPE">
    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="encryptedString"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="double"></xs:enumeration>
    <xs:enumeration value="boolean"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="enum"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
</xs:element>

```

父 元素

[Catalog 元素](#), [Structure 元素](#)

BinaryFormat 元素

XML 表示

```

<xs:element name="BinaryFormat"></xs:element>

```

父 元素

[FileType 元素](#)

Catalog 元素

表 A-7
属性 Catalog

属性	使用	说明	有效值
id	必须的		string
valueColumn	必须的		integer

XML 表示

```
<xs:element name="Catalog">
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:element ref="Attribute"></xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="valueColumn" type="xs:integer" use="required"></xs:attribute>
</xs:element>
```

父 元素

[Catalogs 元素](#)

子 元素

[Attribute 元素](#)

Catalogs 元素

XML 表示

```
<xs:element name="Catalogs">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Catalog"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

父 元素

[CommonObjects 元素](#)

子 元素

[Catalog 元素](#)

ChangeField 元素

表 A-8
属性 ChangeField

属性	使用	说明	有效值
direction	可选的		in out both none partition
directionRef	optional		
fieldRef	必须的		
label	可选的		string
missingValuesRef	optional		
name	必须的		
storage	可选的		unknown integer real string date time timestamp
storageRef	optional		
type	可选的		auto range discrete set orderedSet flag type less
typeRef	optional		

XML 表示

```
<xs:element name="ChangeField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"></xs:element>
      <xs:element ref="Values" minOccurs="0"></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
          <xs:element ref="Range" minOccurs="0"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION"
        minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="real"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

```

    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="time"></xs:enumeration>
    <xs:enumeration value="timestamp"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"></xs:enumeration>
    <xs:enumeration value="range"></xs:enumeration>
    <xs:enumeration value="discrete"></xs:enumeration>
    <xs:enumeration value="set"></xs:enumeration>
    <xs:enumeration value="orderedSet"></xs:enumeration>
    <xs:enumeration value="flag"></xs:enumeration>
    <xs:enumeration value="typeless"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"></xs:attribute>
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:attribute name="storageRef" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
  <xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="directionRef" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
  <xs:attribute name="missingValuesRef" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
</xs:element>

```

父 元素

[ForEach 元素](#), [ModelFields 元素](#)

子 元素

[MissingValues 元素](#), [ModelField 元素](#), [NumericInfo 元素](#), [Range 元素](#), [Values 元素](#)

相关 元素

[AddField 元素](#)

MissingValues 元素

表 A-9
属性 MissingValues

属性	使用	说明	有效值
treatNullAsMissing	optional		boolean
treatWhitespaceAsMissing	optional		boolean

XML 表示

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    <xs:element ref="Range" minOccurs="0"/></xs:element>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
    default="true"/></xs:attribute>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional"
    default="true"/></xs:attribute>
</xs:element>

```

父 元素

[ChangeField](#) 元素

子 元素

[Range](#) 元素, [Values](#) 元素

ModelField 元素

表 A-10
属性 ModelField

属性	使用	说明	有效值
group	可选的		
role	必须的		unknown predictedValue predictedDisplayValue probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	可选的		string
targetField	可选的		string
value	可选的		string

XML 表示

```

<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="predictedValue"/></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"/></xs:enumeration>
    <xs:enumeration value="probability"/></xs:enumeration>
    <xs:enumeration value="residual"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

```

    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
  <xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[ChangeField 元素](#)

CheckBoxControl 元素

表 A-11

属性 CheckBoxControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
invert	optional		boolean
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="CheckBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>

```



```

<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="invert" type="xs:boolean" use="optional"
default="false"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#),
[ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元
素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#),
[PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl
元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元
素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#),
[SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl
元素](#)

CheckBoxGroupControl 元素

表 A-12
属性 CheckBoxGroupControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
layoutByRow	optional		boolean
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
rows	optional		positiveInteger
showLabel	optional		boolean
useSubPanel	optional		boolean

XML 表示

```
<xs:element name="CheckBoxGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [ClientDirectoryChooserControl 元素](#),
[ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元
素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#),
[PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl
元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元
素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#),
[SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl
元素](#)

ClientDirectoryChooserControl 元素

表 A-13

属性 ClientDirectoryChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
mode	必须的		open save import export
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="ClientDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/></xs:enumeration>
    <xs:enumeration value="save"/></xs:enumeration>
    <xs:enumeration value="import"/></xs:enumeration>
    <xs:enumeration value="export"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

父 元素

PropertiesPanel 元素, PropertiesSubPanel 元素

子 元素

Enabled 元素, Layout 元素, Visible 元素

相关 元素

CheckBoxControl 元素, CheckBoxGroupControl 元素, ClientFileChooserControl 元素, DBConnectionChooserControl 元素, DBTableChooserControl 元素, MultiFieldChooserControl 元素, PasswordBoxControl 元素, PropertyControl 元素, RadioButtonGroupControl 元素, ServerDirectoryChooserControl 元素, ServerFileChooserControl 元素, SingleFieldChooserControl 元素, SingleFieldValueChooserControl 元素, SpinnerControl 元素, TableControl 元素, TextAreaControl 元素, TextBoxControl 元素

ClientFileChooserControl 元素

表 A-14

属性 ClientFileChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
mode	必须的		open save import export
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="ClientFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"/></xs:attribute>
</xs:element>
```

```

<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
  <xs:enumeration value="open"></xs:enumeration>
  <xs:enumeration value="save"></xs:enumeration>
  <xs:enumeration value="import"></xs:enumeration>
  <xs:enumeration value="export"></xs:enumeration>
</xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

ComboBoxControl 元素

表 A-15
属性 ComboBoxControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="ComboBoxControl" type="CONTROLLER">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

表 A-16
扩展 类型

类型	说明
ItemChooserControl	

父 元素

[PropertiesPanel](#) 元素, [PropertiesSubPanel](#) 元素

子 元素

[Enabled](#) 元素, [Layout](#) 元素, [Visible](#) 元素

相关 元素

[ActionButton](#) 元素, [ExtensionObjectPanel](#) 元素, [ModelViewerPanel](#) 元素, [SelectorPanel](#) 元素, [StaticText](#) 元素, [SystemControls](#) 元素, [TabbedPanel](#) 元素, [TextBrowserPanel](#) 元素

Command 元素

表 A-17
属性 Command

属性	使用	说明	有效值
path	必须的		

XML 表示

```

<xs:element name="Command">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>

```

父 元素[Run 元素](#)**子 元素**[And 元素](#), [Condition 元素](#), [Not 元素](#), [Or 元素](#)**CommonObjects 元素**

Provides a location for definitions that are global to the extension

表 A-18

属性 CommonObjects

属性	使用	说明	有效值
extensionListenerClass	optional		string

XML 表示

```

<xs:element name="CommonObjects">
  <xs:all>
    <xs:element ref="PropertyTypes" minOccurs="0"></xs:element>
    <xs:element ref="PropertySets" minOccurs="0"></xs:element>
    <xs:element ref="FileFormatTypes" minOccurs="0"></xs:element>
    <xs:element ref="ContainerTypes" minOccurs="0"></xs:element>
    <xs:element ref="Actions" minOccurs="0"></xs:element>
    <xs:element ref="Catalogs" minOccurs="0"></xs:element>
  </xs:all>
  <xs:attribute name="extensionListenerClass" type="xs:string"
    use="optional"></xs:attribute>
</xs:element>

```

父 元素[Extension 元素](#)

子 元素

[Actions](#) 元素, [Catalogs](#) 元素, [ContainerTypes](#) 元素, [FileFormatTypes](#) 元素, [PropertySets](#) 元素, [PropertyTypes](#) 元素

Condition 元素

表 A-19
属性 Condition

属性	使用	说明	有效值
container	optional		string
control	optional		string
expression	optional		
listMode	optional		all any
op	必须的		equals notEquals isEmpty isEmpty lessThan lessOrEquals greaterThan greaterOrEquals equalsIgnoreCase isSubstring startsWith endsWith startsWithIgnoreCase endsWithIgnoreCase isSubstring hasSubstring isSubstringIgnore- Case hasSubstringIgnore- Case in countEquals countLessThan countLessOrEquals countGreaterThan countGreaterOrEquals contains storageEquals typeEquals directionEquals isMeasureDiscrete isMeasureContinuous isMeasureTypeless isMeasureUnknown isStorageString isStorageNumeric isStorageDatetime isStorageUnknown isModelOutput modelOutputRoleE- quals

属性	使用	说明	有效值
			modelOutputTarget-FieldEquals modelOutputHasValue modelOutputTagEquals enumRestriction
property	optional		string
value	optional		

XML 表示

```

<xs:element name="Condition">
  <xs:attribute name="expression" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
  <xs:attribute name="control" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="container" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="op" type="CONDITION-TEST" use="required">
    <xs:enumeration value="equals"></xs:enumeration>
    <xs:enumeration value="notEquals"></xs:enumeration>
    <xs:enumeration value="isEmpty"></xs:enumeration>
    <xs:enumeration value="isNotEmpty"></xs:enumeration>
    <xs:enumeration value="lessThan"></xs:enumeration>
    <xs:enumeration value="lessOrEquals"></xs:enumeration>
    <xs:enumeration value="greaterThan"></xs:enumeration>
    <xs:enumeration value="greaterOrEquals"></xs:enumeration>
    <xs:enumeration value="equalsIgnoreCase"></xs:enumeration>
    <xs:enumeration value="isSubstring"></xs:enumeration>
    <xs:enumeration value="startsWith"></xs:enumeration>
    <xs:enumeration value="endsWith"></xs:enumeration>
    <xs:enumeration value="startsWithIgnoreCase"></xs:enumeration>
    <xs:enumeration value="endsWithIgnoreCase"></xs:enumeration>
    <xs:enumeration value="isSubstring"></xs:enumeration>
    <xs:enumeration value="hasSubstring"></xs:enumeration>
    <xs:enumeration value="isSubstringIgnoreCase"></xs:enumeration>
    <xs:enumeration value="hasSubstringIgnoreCase"></xs:enumeration>
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="countEquals"></xs:enumeration>
    <xs:enumeration value="countLessThan"></xs:enumeration>
    <xs:enumeration value="countLessOrEquals"></xs:enumeration>
    <xs:enumeration value="countGreaterThan"></xs:enumeration>
    <xs:enumeration value="countGreaterOrEquals"></xs:enumeration>
    <xs:enumeration value="contains"></xs:enumeration>
    <xs:enumeration value="storageEquals"></xs:enumeration>
    <xs:enumeration value="typeEquals"></xs:enumeration>
    <xs:enumeration value="directionEquals"></xs:enumeration>
    <xs:enumeration value="isMeasureDiscrete"></xs:enumeration>
    <xs:enumeration value="isMeasureContinuous"></xs:enumeration>
    <xs:enumeration value="isMeasureTypeless"></xs:enumeration>
    <xs:enumeration value="isMeasureUnknown"></xs:enumeration>
    <xs:enumeration value="isStorageString"></xs:enumeration>
    <xs:enumeration value="isStorageNumeric"></xs:enumeration>
    <xs:enumeration value="isStorageDatetime"></xs:enumeration>
    <xs:enumeration value="isStorageUnknown"></xs:enumeration>
    <xs:enumeration value="isModelOutput"></xs:enumeration>
    <xs:enumeration value="modelOutputRoleEquals"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

```

    <xs:enumeration value="modelOutputTargetFieldEquals"></xs:enumeration>
    <xs:enumeration value="modelOutputHasValue"></xs:enumeration>
    <xs:enumeration value="modelOutputTagEquals"></xs:enumeration>
    <xs:enumeration value="enumRestriction"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="value" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="listMode" use="optional" default="all">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="all"></xs:enumeration>
        <xs:enumeration value="any"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>

```

父 元素

And 元素, Command 元素, Constraint 元素, CreateDocument 元素, CreateDocumentOutput 元素, CreateInteractiveDocumentBuilder 元素, CreateInteractiveModelBuilder 元素, CreateModel 元素, CreateModelApplier 元素, CreateModelOutput 元素, Enabled 元素, ExpertSettings 元素, Not 元素, Option 元素, Or 元素, Run 元素, Visible 元素

Constraint 元素

表 A-20
属性 Constraint

属性	使用	说明	有效值
property	必须的		string
singleSelection	optional		boolean

XML 表示

```

<xs:element name="Constraint">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="singleSelection" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
</xs:element>

```

父 元素

AutoModeling 元素

子 元素

[And 元素](#), [Condition 元素](#), [Not 元素](#), [Or 元素](#)

Constructors 元素

XML 表示

```

<xs:element name="Constructors">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CreateModelOutput"></xs:element>
      <xs:element ref="CreateDocumentOutput"></xs:element>
      <xs:element ref="CreateInteractiveModelBuilder"></xs:element>
      <xs:element ref="CreateInteractiveDocumentBuilder"></xs:element>
      <xs:element ref="CreateModelApplier"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

父 元素

[DocumentOutput 元素](#), [Execution 元素](#), [InteractiveDocumentBuilder 元素](#), [InteractiveModelBuilder 元素](#), [ModelOutput 元素](#), [Node 元素](#)

子 元素

[CreateDocumentOutput 元素](#), [CreateInteractiveDocumentBuilder 元素](#), [CreateInteractiveModelBuilder 元素](#), [CreateModelApplier 元素](#), [CreateModelOutput 元素](#)

Container 元素

表 A-21
属性 Container

属性	使用	说明	有效值
name	必须的		string
type	optional		string

XML 表示

```

<xs:element name="Container">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Containers 元素](#), [Containers 元素](#), [Containers 元素](#), [Containers 元素](#), [Containers 元素](#)

ContainerFile 元素

表 A-22
属性 ContainerFile

属性	使用	说明	有效值
container	optional		
containerType	optional		
path	必须的		

XML 表示

```
<xs:element name="ContainerFile" type="SERVER-CONTAINER-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:attribute name="container" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="containerType" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
</xs:element>
```

父 元素

[InputFiles 元素](#), [OutputFiles 元素](#)

ContainerTypes 元素

XML 表示

```
<xs:element name="ContainerTypes">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="DocumentType"></xs:element>
      <xs:element ref="ModelType"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

父 元素

[CommonObjects 元素](#)

子 元素

[DocumentType 元素](#), [ModelType 元素](#)

Controls 元素

XML 表示

```
<xs:element name="Controls">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Menu"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

        <xs:element ref="MenuItem"></xs:element>
        <xs:element ref="ToolBarItem"></xs:element>
    </xs:choice>
</xs:sequence>
</xs:element>

```

父 元素

[UserInterface](#) 元素

子 元素

[Menu](#) 元素, [MenuItem](#) 元素, [ToolBarItem](#) 元素

CreateDocument 元素

表 A-23
属性 CreateDocument

属性	使用	说明	有效值
sourceFile	必须的		string
target	必须的		string
type	optional		string

XML 表示

```

<xs:element name="CreateDocument">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"></xs:element>
      <xs:element ref="And"></xs:element>
      <xs:element ref="Or"></xs:element>
      <xs:element ref="Not"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[CreateDocumentOutput](#) 元素, [CreateInteractiveDocumentBuilder](#) 元素,
[CreateInteractiveModelBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#)
元素

子 元素

[And](#) 元素, [Condition](#) 元素, [Not](#) 元素, [Or](#) 元素

相关 元素

[CreateModel](#) 元素

CreateDocumentOutput 元素

表 A-24
属性 CreateDocumentOutput

属性	使用	说明	有效值
type	必须的		string

XML 表示

```
<xs:element name="CreateDocumentOutput">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[Constructors](#) 元素

子 元素

[And](#) 元素, [Condition](#) 元素, [CreateDocument](#) 元素, [CreateModel](#) 元素, [Not](#) 元素, [Or](#) 元素, [SetContainer](#) 元素, [SetProperty](#) 元素

相关 元素

[CreateInteractiveDocumentBuilder](#) 元素, [CreateInteractiveModelBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#) 元素

CreateInteractiveDocumentBuilder 元素

表 A-25
属性 CreateInteractiveDocumentBuilder

属性	使用	说明	有效值
type	必须的		string

XML 表示

```

<xs:element name="CreateInteractiveDocumentBuilder">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[Constructors](#) 元素

子 元素

[And](#) 元素, [Condition](#) 元素, [CreateDocument](#) 元素, [CreateModel](#) 元素, [Not](#) 元素, [Or](#) 元素, [SetContainer](#) 元素, [SetProperty](#) 元素

相关 元素

[CreateDocumentOutput](#) 元素, [CreateInteractiveModelBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#) 元素

CreateInteractiveModelBuilder 元素

表 A-26
属性 CreateInteractiveModelBuilder

属性	使用	说明	有效值
type	必须的		string

XML 表示

```

<xs:element name="CreateInteractiveModelBuilder">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Not"></xs:element>
    </xs:choice>
</xs:group>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
    </xs:choice>
</xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[Constructors](#) 元素

子 元素

[And](#) 元素, [Condition](#) 元素, [CreateDocument](#) 元素, [CreateModel](#) 元素, [Not](#) 元素, [Or](#) 元素, [SetContainer](#) 元素, [SetProperty](#) 元素

相关 元素

[CreateDocumentOutput](#) 元素, [CreateInteractiveDocumentBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#) 元素

CreateModel 元素

表 A-27
属性 CreateModel

属性	使用	说明	有效值
signatureFile	optional		string
sourceFile	必须的		string
target	必须的		string
type	optional		string

XML 表示

```

<xs:element name="CreateModel">
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
        <xs:choice>
            <xs:element ref="Condition"></xs:element>
            <xs:element ref="And"></xs:element>
            <xs:element ref="Or"></xs:element>
            <xs:element ref="Not"></xs:element>
        </xs:choice>
    </xs:group>
    <xs:attribute name="sourceFile" type="xs:string" use="required"></xs:attribute>
    <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
    <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>

```



```

    <xs:sequence>
      <xs:element name="ModelDetail" maxOccurs="unbounded"/></xs:element>
    </xs:sequence>
    <xs:attribute name="signatureFile" type="xs:string" use="optional"/></xs:attribute>
  </xs:element>

```

父 元素

[CreateDocumentOutput](#) 元素, [CreateInteractiveDocumentBuilder](#) 元素, [CreateInteractiveModelBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#) 元素

子 元素

[And](#) 元素, [Condition](#) 元素, [ModelDetail](#) 元素, [Not](#) 元素, [Or](#) 元素

相关 元素

[CreateDocument](#) 元素

ModelDetail 元素

表 A-28
属性 ModelDetail

属性	使用	说明	有效值
algorithm	必须的		string

XML 表示

```

<xs:element name="ModelDetail" maxOccurs="unbounded">
  <xs:attribute name="algorithm" type="xs:string" use="required"/></xs:attribute>
</xs:element>

```

父 元素

[CreateModel](#) 元素

CreateModelApplier 元素

表 A-29
属性 CreateModelApplier

属性	使用	说明	有效值
type	必须的		string

XML 表示

```

<xs:element name="CreateModelApplier">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/></xs:element>
        <xs:element ref="And"/></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
    </xs:choice>
</xs:group>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
    </xs:choice>
</xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

Constructors 元素

子 元素

And 元素, Condition 元素, CreateDocument 元素, CreateModel 元素, Not 元素, Or 元素, SetContainer 元素, SetProperty 元素

相关 元素

CreateDocumentOutput 元素, CreateInteractiveDocumentBuilder 元素, CreateInteractiveModelBuilder 元素, CreateModelOutput 元素

CreateModelOutput 元素

表 A-30
属性 CreateModelOutput

属性	使用	说明	有效值
type	必须的		string

XML 表示

```

<xs:element name="CreateModelOutput">
    <xs:sequence>
        <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
            <xs:choice>
                <xs:element ref="Condition"></xs:element>
                <xs:element ref="And"></xs:element>
                <xs:element ref="Or"></xs:element>
                <xs:element ref="Not"></xs:element>
            </xs:choice>
        </xs:group>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:choice>
                <xs:element ref="SetProperty"></xs:element>
                <xs:element ref="SetContainer"></xs:element>
                <xs:element ref="CreateModel"></xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
</xs:element>

```

```

        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[Constructors 元素](#)

子 元素

[And 元素](#), [Condition 元素](#), [CreateDocument 元素](#), [CreateModel 元素](#), [Not 元素](#), [Or 元素](#), [SetContainer 元素](#), [SetProperty 元素](#)

相关 元素

[CreateDocumentOutput 元素](#), [CreateInteractiveDocumentBuilder 元素](#),
[CreateInteractiveModelBuilder 元素](#), [CreateModelApplier 元素](#)

DatabaseConnectionValue 元素

A value specifying the details of a database connection.

表 A-31
属性 DatabaseConnectionValue

属性	使用	说明	有效值
connectionType	必须的		string
datasourceName	必须的		string
password	必须的		string
userName	必须的		string

XML 表示

```

<xs:element name="DatabaseConnectionValue" type="DATABASE-CONNECTION-VALUE">
  <xs:attribute name="connectionType" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="datasourceName" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="userName" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="password" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[ListValue 元素](#), [Attribute 元素](#), [ListValue 元素](#), [Parameter 元素](#), [Attribute 元素](#), [ListValue 元素](#)

DataFile 元素

表 A-32
属性 DataFile

属性	使用	说明	有效值
path	必须的		

XML 表示

```
<xs:element name="DataFile" type="SERVER-DATA-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:choice>
    <xs:element ref="DelimitedDataFormat"></xs:element>
  </xs:choice>
</xs:element>
```

父 元素

[InputFiles 元素](#), [OutputFiles 元素](#)

子 元素

[DelimitedDataFormat 元素](#)

DataFormat 元素

XML 表示

```
<xs:element name="DataFormat">
  <xs:group ref="DATA-FORMAT-TYPE">
    <xs:choice>
      <xs:element ref="DelimitedDataFormat"></xs:element>
      <xs:element ref="SPSSDataFormat"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

父 元素

[FileFormatType 元素](#)

子 元素

[DelimitedDataFormat 元素](#), [SPSSDataFormat 元素](#)

DataModel 元素

The data model coming into or out of a node. An input/output data model is a set of Fields1.

XML 表示

```

<xs:element name="DataModel" type="DATA-MODEL">
  <xs:sequence>
    <xs:element name="FieldFormats" type="FIELD-FORMATS" minOccurs="0">
      <xs:sequence>
        <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION"
          minOccurs="0" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="FieldGroups" type="FIELD-GROUPS" minOccurs="0">
      <xs:sequence>
        <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0"
          maxOccurs="unbounded">
          <xs:sequence>
            <xs:element name="FieldName"></xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="Fields" type="FIELDS">
      <xs:sequence>
        <xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
          <xs:group ref="FIELD-CONTENT">
            <xs:sequence>
              <xs:element ref="DisplayLabel"></xs:element>
              <xs:choice minOccurs="0">
                <xs:element ref="Range"></xs:element>
                <xs:element ref="Values"></xs:element>
              </xs:choice>
              <xs:element ref="MissingValues"></xs:element>
            </xs:sequence>
          </xs:group>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

子 元素

[FieldFormats 元素](#), [FieldGroups 元素](#), [Fields 元素](#)

FieldFormats 元素

Defines the default field formats. Field formats are used when displaying values in output such as the general format (standard number, scientific or currency formats), number of decimal places to display, decimal separator etc. Currently field formats are only used for numeric fields although this may change in future versions.

表 A-33
属性 FieldFormats

属性	使用	说明	有效值
count	可选的		nonNegativeInteger
defaultNumberFormat	必须的		string

XML 表示

```
<xs:element name="FieldFormats" type="FIELD-FORMATS" minOccurs="0">
  <xs:sequence>
    <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="defaultNumberFormat" type="xs:string"
    use="required"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

父 元素

[DataModel 元素](#)

子 元素

[NumberFormat 元素](#)

NumberFormat 元素

Defines format information for a numeric field.

表 A-34
属性 NumberFormat

属性	使用	说明	有效值
decimalPlaces	必须的		nonNegativeInteger
decimalSymbol	必须的		period comma
formatType	必须的		standard scientific currency
groupingSymbol	必须的		none period comma space
name	必须的		string

XML 表示

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
  maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
```

```

    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger"
use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

父 元素

[FieldFormats](#) 元素

FieldGroups 元素

Defines the field groups. Field groups are used to associate related fields. For example, a survey question that asks a respondent to select which locations they have visited from a set of options will be represented as a set of flag fields. A field group may be used to identify which fields are associated with that survey question.

表 A-35
属性 FieldGroups

属性	使用	说明	有效值
count	可选的		nonNegativeInteger

XML 表示

```

<xs:element name="FieldGroups" type="FIELD-GROUPS" minOccurs="0">
  <xs:sequence>
    <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0"
maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="FieldName"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

父 元素

[DataModel](#) 元素

子 元素

[FieldGroup 元素](#)

FieldGroup 元素

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents “true”.

表 A-36
属性 FieldGroup

属性	使用	说明	有效值
count	可选的		nonNegativeInteger
countedValue	可选的		string
displayLabel	可选的		string
groupType	必须的		fieldGroup multiCategorySet multiDichotomySet
name	必须的		

XML 表示

```
<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0"
maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="FieldName"></xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
  <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
    <xs:enumeration value="fieldGroup"></xs:enumeration>
    <xs:enumeration value="multiCategorySet"></xs:enumeration>
    <xs:enumeration value="multiDichotomySet"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="countedValue" type="xs:string"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

父 元素

[FieldGroups 元素](#)

子 元素

[FieldName 元素](#)

FieldName 元素

表 A-37

属性 FieldName

属性	使用	说明	有效值
name	必须的		

XML 表示

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"/></xs:attribute>
</xs:element>
```

父 元素

[FieldGroup 元素](#)

Fields 元素

表 A-38

属性 Fields

属性	使用	说明	有效值
count	可选的		nonNegativeInteger

XML 表示

```
<xs:element name="Fields" type="FIELDS">
  <xs:sequence>
    <xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="FIELD-CONTENT">
        <xs:sequence>
          <xs:element ref="DisplayLabel"/></xs:element>
          <xs:choice minOccurs="0">
            <xs:element ref="Range"/></xs:element>
            <xs:element ref="Values"/></xs:element>
          </xs:choice>
          <xs:element ref="MissingValues"/></xs:element>
        </xs:sequence>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"/></xs:attribute>
</xs:element>
```

父 元素

[DataModel 元素](#)

子 元素

[Field 元素](#)

Field 元素

表 A-39
属性 Field

属性	使用	说明	有效值
direction	可选的		in out both none partition
displayLabel	可选的		string
name	必须的		string
storage	可选的		unknown integer real string date time timestamp
type	可选的		auto range discrete set orderedSet flag typeless

XML 表示

```

<xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
  <xs:group ref="FIELD-CONTENT">
    <xs:sequence>
      <xs:element ref="DisplayLabel"/>
      <xs:choice minOccurs="0">
        <xs:element ref="Range"/>
        <xs:element ref="Values"/>
      </xs:choice>
      <xs:element ref="MissingValues"/>
    </xs:sequence>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="FIELD-TYPE" default="auto">
    <xs:enumeration value="auto"/>
    <xs:enumeration value="range"/>
    <xs:enumeration value="discrete"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="orderedSet"/>
    <xs:enumeration value="flag"/>
    <xs:enumeration value="typeless"/>
  </xs:attribute>
  <xs:attribute name="storage" type="FIELD-STORAGE" default="unknown">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="real"/>
    <xs:enumeration value="string"/>
    <xs:enumeration value="date"/>
  </xs:attribute>

```

```

    <xs:enumeration value="time"></xs:enumeration>
    <xs:enumeration value="timestamp"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION" default="in">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[Fields 元素](#)

子 元素

[DisplayLabel 元素](#), [MissingValues 元素](#), [Range 元素](#), [Values 元素](#)

DBCConnectionChooserControl 元素

表 A-40

属性 DBCConnectionChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="DBCConnectionChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>

```

```

<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

DBTableChooserControl 元素

表 A-41

属性 DBTableChooserControl

属性	使用	说明	有效值
connectionProperty	必须的		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="DBTableChooserControl">
  <xs:sequence>
    <xs:choice>

```

```

    <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
    <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
    <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:choice>
</xs:sequence>
<xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="showLabel" type="xs:boolean" use="optional"
default="true"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="connectionProperty" type="xs:string"
use="required"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

DefaultValue 元素

XML 表示

```

<xs:element name="DefaultValue">
  <xs:choice>
    <xs:element name="ServerTempFile"></xs:element>
    <xs:element name="ServerTempDir"></xs:element>
    <xs:element name="Identifier"></xs:element>
  </xs:choice>
</xs:element>

```

父 元素

[Property](#) 元素, [PropertyType](#) 元素

子 元素

[Identifier](#) 元素, [ServerTempDir](#) 元素, [ServerTempFile](#) 元素

ServerTempFile 元素

表 A-42

属性 ServerTempFile

属性	使用	说明	有效值
basename	必须的		

XML 表示

```
<xs:element name="ServerTempFile">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

父 元素

[DefaultValue](#) 元素

ServerTempDir 元素

表 A-43

属性 ServerTempDir

属性	使用	说明	有效值
basename	必须的		

XML 表示

```
<xs:element name="ServerTempDir">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

父 元素

[DefaultValue](#) 元素

Identifier 元素

表 A-44

属性 Identifier

属性	使用	说明	有效值
basename	必须的		

XML 表示

```
<xs:element name="Identifier">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"/></xs:attribute>
</xs:element>
```

父 元素

[DefaultValue](#) 元素

DelimitedDataFormat 元素

表 A-45
属性 DelimitedDataFormat

属性	使用	说明	有效值
delimiter	optional		tab comma semicolon colon verticalBar other
eol	optional		cr cr lf lf other
includeFieldNames	optional		boolean
otherDelimiter	optional		string
otherEol	optional		string
quoteStrings	optional		boolean
stringQuote	optional		string

XML 表示

```
<xs:element name="DelimitedDataFormat">
  <xs:attribute name="delimiter" use="optional" default="tab">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="tab"/></xs:enumeration>
        <xs:enumeration value="comma"/></xs:enumeration>
        <xs:enumeration value="semicolon"/></xs:enumeration>
        <xs:enumeration value="colon"/></xs:enumeration>
        <xs:enumeration value="verticalBar"/></xs:enumeration>
        <xs:enumeration value="other"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="otherDelimiter" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="eol" use="optional" default="cr">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="cr"/></xs:enumeration>
        <xs:enumeration value="cr lf"/></xs:enumeration>
        <xs:enumeration value="lf"/></xs:enumeration>
        <xs:enumeration value="other"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```

```

        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="otherEol" type="xs:string" use="optional"></xs:attribute>
    <xs:attribute name="includeFieldNames" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
    <xs:attribute name="quoteStrings" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
    <xs:attribute name="stringQuote" type="xs:string" use="optional"
    default=""></xs:attribute>
  </xs:element>

```

父 元素

[DataFile 元素](#), [DataFormat 元素](#)

DisplayLabel 元素

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

表 A-46

属性 DisplayLabel

属性	使用	说明	有效值
lang	可选的		NMTOKEN
value	必须的		string

XML 表示

```

<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>

```

父 元素

[Field 元素](#)

DocumentBuilder 元素

XML 表示

```

<xs:element name="DocumentBuilder">
  <xs:sequence>
    <xs:element name="DocumentGeneration"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[Node 元素](#)

子 元素

[DocumentGeneration 元素](#)

DocumentGeneration 元素

表 A-47

属性 DocumentGeneration

属性	使用	说明	有效值
controlsId	必须的		string

XML 表示

```
<xs:element name="DocumentGeneration">
  <xs:attribute name="controlsId" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

父 元素

[DocumentBuilder 元素](#)

相关 元素

[ModelingFields 元素](#), [ModelGeneration 元素](#), [ModelEvaluation 元素](#)

DocumentOutput 元素

表 A-48

属性 DocumentOutput

属性	使用	说明	有效值
deprecatedScriptNames	optional		string
id	必须的		string
scriptName	optional		string

XML 表示

```
<xs:element name="DocumentOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/></xs:element>
      <xs:element ref="Constructors" minOccurs="0"/></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/></xs:attribute>
```

```
<xs:attribute name="deprecatedScriptNames" type="xs:string"
  use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Extension 元素](#)

子 元素

[Constructors 元素](#), [Containers 元素](#), [ModelProvider 元素](#), [Properties 元素](#),
[UserInterface 元素](#)

相关 元素

[InteractiveDocumentBuilder 元素](#), [InteractiveModelBuilder 元素](#), [ModelOutput 元素](#), [Node 元素](#)

Containers 元素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[DocumentOutput 元素](#)

子 元素

[Container 元素](#)

DocumentType 元素

Defines a new document type

表 A-49

属性 DocumentType

属性	使用	说明	有效值
format	必须的		utf8 binary
id	必须的		string
type	optional		unknown rowSet report graph

XML 表示

```

<xs:element name="DocumentType">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/></xs:enumeration>
        <xs:enumeration value="binary"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="type" type="DOCUMENT-TYPE" use="optional">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="rowSet"/></xs:enumeration>
    <xs:enumeration value="report"/></xs:enumeration>
    <xs:enumeration value="graph"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

父 元素

[ContainerTypes 元素](#)

相关 元素

[ModelType 元素](#)

Enabled 元素

XML 表示

```

<xs:element name="Enabled">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/></xs:element>
        <xs:element ref="And"/></xs:element>
        <xs:element ref="Or"/></xs:element>
        <xs:element ref="Not"/></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

父 元素

[ActionButton 元素](#), [CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [ComboBoxControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [ExtensionObjectPanel 元素](#), [ModelViewerPanel 元素](#), [MultiFieldChooserControl 元素](#), [MultiItemChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [PropertyControl 元素](#),

RadioButtonGroupControl 元素, SelectorPanel 元素, ServerDirectoryChooserControl 元素, ServerFileChooserControl 元素, SingleFieldChooserControl 元素, SingleFieldValueChooserControl 元素, SingleItemChooserControl 元素, SpinnerControl 元素, StaticText 元素, SystemControls 元素, TabbedPanel 元素, TableControl 元素, TextAreaControl 元素, TextBoxControl 元素, TextBrowserPanel 元素, ItemChooserControl 类型

子 元素

And 元素, Condition 元素, Not 元素, Or 元素

Enumeration 元素

XML 表示

```
<xs:element name="Enumeration">
  <xs:sequence>
    <xs:element name="Enum" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

PropertyType 元素

子 元素

Enum 元素

Enum 元素

表 A-50
属性 Enum

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
imagePath	optional		string
imagePathKey	optional		string
label	必须的		string
labelKey	optional		string
value	必须的		string

XML 表示

```
<xs:element name="Enum" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
```

```

    <xs:attribute name="imagePath" type="xs:string" use="optional"></xs:attribute>
    <xs:attribute name="imagePathKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Enumeration 元素](#)

ErrorDetail 元素

Supplementary information about an error or other condition.

XML 表示

```

<xs:element name="ErrorDetail" type="ERROR-DETAIL">
  <xs:sequence>
    <xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0"
      maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="Message" type="DIAGNOSTIC-MESSAGE"
          minOccurs="0"></xs:element>
        <xs:element name="Parameter" type="xs:string" minOccurs="0"
          maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

子 元素

[Diagnostic 元素](#)

Diagnostic 元素

表 A-51
属性 Diagnostic

属性	使用	说明	有效值
code	必须的		integer
severity	可选的		unknown information warning error fatal
source	可选的		string
subCode	可选的		integer

XML 表示

```

<xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:element>

```

```

    <xs:element name="Parameter" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="subCode" type="xs:integer" default="0"></xs:attribute>
  <xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="information"></xs:enumeration>
    <xs:enumeration value="warning"></xs:enumeration>
    <xs:enumeration value="error"></xs:enumeration>
    <xs:enumeration value="fatal"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="source" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[ErrorDetail 元素](#)

子 元素

[Message 元素](#), [Parameter 元素](#)

Message 元素

表 A-52
属性 Message

属性	使用	说明	有效值
lang	可选的		NMTOKEN

XML 表示

```

<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>

```

父 元素

[Diagnostic 元素](#)

Parameter 元素

XML 表示

```

<xs:element name="Parameter" type="xs:string" minOccurs="0"
  maxOccurs="unbounded"></xs:element>

```

父 元素

[Diagnostic 元素](#)

Executable 元素

XML 表示

```
<xs:element name="Executable">
  <xs:sequence>
    <xs:element ref="Run" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[Execution 元素](#)

子 元素

[Run 元素](#)

Execution 元素

XML 表示

```
<xs:element name="Execution">
  <xs:sequence>
    <xs:element ref="Properties" minOccurs="0"></xs:element>
    <xs:element ref="InputFiles"></xs:element>
    <xs:element ref="OutputFiles"></xs:element>
    <xs:choice>
      <xs:element ref="Executable"></xs:element>
      <xs:element ref="Module"></xs:element>
    </xs:choice>
    <xs:element ref="Constructors" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[Node 元素](#)

子 元素

[Constructors 元素](#), [Executable 元素](#), [InputFiles 元素](#), [Module 元素](#), [OutputFiles 元素](#), [Properties 元素](#)

Extension 元素

Defines the top-level extension container.

表 A-53
属性 Extension

属性	使用	说明	有效值
debug	optional		boolean
version	必须的		string

XML 表示

```

<xs:element name="Extension">
  <xs:sequence>
    <xs:element ref="ExtensionDetails"></xs:element>
    <xs:element ref="Resources"></xs:element>
    <xs:element ref="License" minOccurs="0"></xs:element>
    <xs:element ref="CommonObjects"></xs:element>
    <xs:element ref="UserInterface" minOccurs="0"></xs:element>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="Node"></xs:element>
        <xs:element ref="ModelOutput"></xs:element>
        <xs:element ref="DocumentOutput"></xs:element>
        <xs:element ref="InteractiveModelBuilder"></xs:element>
        <xs:element ref="InteractiveDocumentBuilder"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="debug" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
</xs:element>

```

子 元素

[CommonObjects](#) 元素, [DocumentOutput](#) 元素, [ExtensionDetails](#) 元素,
[InteractiveDocumentBuilder](#) 元素, [InteractiveModelBuilder](#) 元素, [License](#) 元素,
[ModelOutput](#) 元素, [Node](#) 元素, [Resources](#) 元素, [UserInterface](#) 元素

ExtensionDetails 元素

Defines information about the extension such as the extension id, the extension provider and version information.

表 A-54
属性 ExtensionDetails

属性	使用	说明	有效值
copyright	optional		string
description	optional		string
id	必须的		string
label	必须的		string
provider	optional		string
providerTag	必须的		string
version	可选的		string

XML 表示

```
<xs:element name="ExtensionDetails">
  <xs:attribute name="providerTag" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="version" type="xs:string"></xs:attribute>
  <xs:attribute name="provider" type="xs:string" use="optional"
    default="(unknown)"></xs:attribute>
  <xs:attribute name="copyright" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Extension 元素](#)

ExtensionObjectPanel 元素

表 A-55

属性 ExtensionObjectPanel

属性	使用	说明	有效值
id	optional		string
panelClass	必须的		string

XML 表示

```
<xs:element name="ExtensionObjectPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="panelClass" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [Tab 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ModelViewerPanel 元素](#), [SelectorPanel 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#), [TextBrowserPanel 元素](#)

Field 元素

表 A-56
属性 Field

属性	使用	说明	有效值
direction	可选的		in out both none partition
label	可选的		string
name	必须的		
storage	可选的		unknown integer real string date time timestamp
type	可选的		auto range discrete set orderedSet flag typeless

XML 表示

```

<xs:element name="Field" type="FIELD-DECLARATION">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/></xs:element>
      <xs:element ref="Values" minOccurs="0"/></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"/></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/></xs:element>
          <xs:element ref="Range" minOccurs="0"/></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION"
        minOccurs="0"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-NAME" use="required"/></xs:attribute>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="real"/></xs:enumeration>
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="time"/></xs:enumeration>
    <xs:enumeration value="timestamp"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">

```

```

    <xs:enumeration value="auto"></xs:enumeration>
    <xs:enumeration value="range"></xs:enumeration>
    <xs:enumeration value="discrete"></xs:enumeration>
    <xs:enumeration value="set"></xs:enumeration>
    <xs:enumeration value="orderedSet"></xs:enumeration>
    <xs:enumeration value="flag"></xs:enumeration>
    <xs:enumeration value="typeless"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"></xs:attribute>
</xs:element>

```

子 元素

[MissingValues](#) 元素, [ModelField](#) 元素, [NumericInfo](#) 元素, [Range](#) 元素, [Values](#) 元素

MissingValues 元素

表 A-57
属性 MissingValues

属性	使用	说明	有效值
treatNullAsMissing	optional		boolean
treatWhitespaceAsMissing	optional		boolean

XML 表示

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="Range" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
</xs:element>

```

父 元素

[Field](#) 元素

子 元素

[Range](#) 元素, [Values](#) 元素

ModelField 元素

表 A-58
属性 ModelField

属性	使用	说明	有效值
group	可选的		
role	必须的		unknown predictedValue predictedDisplay- Value probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	可选的		string
targetField	可选的		string
value	可选的		string

XML 表示

```

<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="predictedValue"></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
    <xs:enumeration value="probability"></xs:enumeration>
    <xs:enumeration value="residual"></xs:enumeration>
    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
  <xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[Field 元素](#)

FieldFormats 元素

Defines the default field formats. Field formats are used when displaying values in output such as the general format (standard number, scientific or currency formats), number of decimal places to display, decimal separator etc. Currently field formats are only used for numeric fields although this may change in future versions.

表 A-59
属性 FieldFormats

属性	使用	说明	有效值
count	可选的		nonNegativeInteger
defaultNumberFormat	必须的		string

XML 表示

```
<xs:element name="FieldFormats" type="FIELD-FORMATS">
  <xs:sequence>
    <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="defaultNumberFormat" type="xs:string"
    use="required"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

子 元素

[NumberFormat 元素](#)

NumberFormat 元素

Defines format information for a numeric field.

表 A-60
属性 NumberFormat

属性	使用	说明	有效值
decimalPlaces	必须的		nonNegativeInteger
decimalSymbol	必须的		period comma
formatType	必须的		standard scientific currency
groupingSymbol	必须的		none period comma space
name	必须的		string

XML 表示

```

<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger"
use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

父 元素

[FieldFormats](#) 元素

FieldGroup 元素

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents “true”.

表 A-61

属性 FieldGroup

属性	使用	说明	有效值
count	可选的		nonNegativeInteger
countedValue	可选的		string
displayLabel	可选的		string
groupType	必须的		fieldGroup multiCategorySet multiDichotomySet
name	必须的		

XML 表示

```

<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION">
  <xs:sequence>
    <xs:element name="FieldName"></xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"></xs:attribute>

```

```

<xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
<xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
  <xs:enumeration value="fieldGroup"></xs:enumeration>
  <xs:enumeration value="multiCategorySet"></xs:enumeration>
  <xs:enumeration value="multiDichotomySet"></xs:enumeration>
</xs:attribute>
<xs:attribute name="countedValue" type="xs:string"></xs:attribute>
<xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

子 元素

[FieldName 元素](#)

FieldName 元素

表 A-62
属性 FieldName

属性	使用	说明	有效值
name	必须的		

XML 表示

```

<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>

```

父 元素

[FieldGroup 元素](#)

FieldGroups 元素

Defines the field groups. Field groups are used to associate related fields. For example, a survey question that asks a respondent to select which locations they have visited from a set of options will be represented as a set of flag fields. A field group may be used to identify which fields are associated with that survey question.

表 A-63
属性 FieldGroups

属性	使用	说明	有效值
count	可选的		nonNegativeInteger

XML 表示

```

<xs:element name="FieldGroups" type="FIELD-GROUPS">
  <xs:sequence>
    <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0"
      maxOccurs="unbounded">
      <xs:sequence>

```

```

        <xs:element name="FieldName"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

子 元素

[FieldGroup 元素](#)

FieldGroup 元素

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents “true”.

表 A-64
属性 FieldGroup

属性	使用	说明	有效值
count	可选的		nonNegativeInteger
countedValue	可选的		string
displayLabel	可选的		string
groupType	必须的		fieldGroup multiCategorySet multiDichotomySet
name	必须的		

XML 表示

```

<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0"
maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="FieldName"></xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
  <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
    <xs:enumeration value="fieldGroup"></xs:enumeration>
    <xs:enumeration value="multiCategorySet"></xs:enumeration>
    <xs:enumeration value="multiDichotomySet"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="countedValue" type="xs:string"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

父 元素

[FieldGroups 元素](#)

子 元素[FieldName 元素](#)**FieldName 元素**

表 A-65

属性 FieldName

属性	使用	说明	有效值
name	必须的		

XML 表示

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

父 元素[FieldGroup 元素](#)**FileFormatType 元素**

表 A-66

属性 FileFormatType

属性	使用	说明	有效值
name	optional		

XML 表示

```
<xs:element name="FileFormatType">
  <xs:sequence>
    <xs:group ref="FILE-FORMAT">
      <xs:choice>
        <xs:element ref="UTF8Format"></xs:element>
        <xs:element ref="BinaryFormat"></xs:element>
        <xs:element ref="DataFormat"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="name" type="EVALUATED-STRING" use="optional"></xs:attribute>
</xs:element>
```

父 元素[FileFormatTypes 元素](#)**子 元素**[BinaryFormat 元素](#), [DataFormat 元素](#), [UTF8Format 元素](#)

FileFormatTypes 元素

XML 表示

```
<xs:element name="FileFormatTypes">
  <xs:sequence>
    <xs:element ref="FileFormatType" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[CommonObjects 元素](#)

子 元素

[FileFormatType 元素](#)

ForEach 元素

表 A-67
属性 ForEach

属性	使用	说明	有效值
container	optional		string
from	optional		string
inFields	optional		string
inFieldValues	optional		string
inProperty	optional		string
step	optional		string
to	optional		string
var	必须的		string

XML 表示

```
<xs:element name="ForEach">
  <xs:sequence maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"/></xs:element>
        <xs:element ref="AddField"/></xs:element>
        <xs:element ref="ChangeField"/></xs:element>
        <xs:element ref="RemoveField"/></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="var" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="inProperty" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="inFields" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="inFieldValues" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="from" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="to" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="step" type="xs:string" use="optional"/></xs:attribute>
```

```
<xs:attribute name="container" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[ModelFields 元素](#)

子 元素

[AddField 元素](#), [ChangeField 元素](#), [ForEach 元素](#), [RemoveField 元素](#)

Icon 元素

表 A-68
属性 Icon

属性	使用	说明	有效值
imagePath	必须的		string
resourceID	optional		string
type	必须的		standardNode smallNode standardWindow

XML 表示

```
<xs:element name="Icon">
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="standardNode"></xs:enumeration>
        <xs:enumeration value="smallNode"></xs:enumeration>
        <xs:enumeration value="standardWindow"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="imagePath" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="resourceID" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Icons 元素](#), [Palette 元素](#)

Icons 元素

XML 表示

```
<xs:element name="Icons">
  <xs:sequence>
    <xs:element ref="Icon" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素[UserInterface 元素](#)**子 元素**[Icon 元素](#)**InputFiles 元素****XML 表示**

```
<xs:element name="InputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"></xs:element>
      <xs:element ref="ContainerFile" minOccurs="0"
        maxOccurs="unbounded"></xs:element>
    </xs:sequence>
  </xs:group>
</xs:element>
```

父 元素[Execution 元素](#), [Module 元素](#)**子 元素**[ContainerFile 元素](#), [DataFile 元素](#)**InteractiveDocumentBuilder 元素**

表 A-69

属性 InteractiveDocumentBuilder

属性	使用	说明	有效值
deprecatedScriptNames	optional		string
id	必须的		string
scriptName	optional		string

XML 表示

```
<xs:element name="InteractiveDocumentBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

        <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="deprecatedScriptNames" type="xs:string"
use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Extension 元素](#)

子 元素

[Constructors 元素](#), [Containers 元素](#), [ModelProvider 元素](#), [Properties 元素](#), [UserInterface 元素](#)

相关 元素

[DocumentOutput 元素](#), [InteractiveModelBuilder 元素](#), [ModelOutput 元素](#), [Node 元素](#)

Containers 元素

XML 表示

```

<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[InteractiveDocumentBuilder 元素](#)

子 元素

[Container 元素](#)

InteractiveModelBuilder 元素

表 A-70
属性 InteractiveModelBuilder

属性	使用	说明	有效值
deprecatedScriptNames	optional		string
id	必须的		string
scriptName	optional		string

XML 表示

```
<xs:element name="InteractiveModelBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string"
    use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Extension 元素](#)

子 元素

[Constructors 元素](#), [Containers 元素](#), [ModelProvider 元素](#), [Properties 元素](#), [UserInterface 元素](#)

相关 元素

[DocumentOutput 元素](#), [InteractiveDocumentBuilder 元素](#), [ModelOutput 元素](#), [Node 元素](#)

Containers 元素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[InteractiveModelBuilder 元素](#)

子 元素

[Container 元素](#)

Layout 元素

表 A-71
属性 Layout

属性	使用	说明	有效值
anchor	optional		north northeast east southeast south southwest west northwest center
columnWeight	optional		double
fill	optional		horizontal vertical both none
gridColumn	optional		nonNegativeInteger
gridHeight	optional		nonNegativeInteger
gridRow	optional		nonNegativeInteger
gridWidth	optional		nonNegativeInteger
leftIndent	optional		nonNegativeInteger
rowIncrement	optional		nonNegativeInteger
rowWeight	optional		double

XML 表示

```

<xs:element name="Layout">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Cell"></xs:element>
  </xs:sequence>
  <xs:attribute name="gridRow" type="xs:nonNegativeInteger"
    use="optional"></xs:attribute>
  <xs:attribute name="gridColumn" type="xs:nonNegativeInteger"
    use="optional"></xs:attribute>
  <xs:attribute name="rowIncrement" type="xs:nonNegativeInteger"
    use="optional"></xs:attribute>
  <xs:attribute name="gridWidth" type="xs:nonNegativeInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="gridHeight" type="xs:nonNegativeInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="rowWeight" type="xs:double" use="optional"></xs:attribute>
  <xs:attribute name="columnWeight" type="xs:double" use="optional"></xs:attribute>
  <xs:attribute name="fill" type="UI-COMPONENT-FILL" use="optional" default="none">
    <xs:enumeration value="horizontal"></xs:enumeration>
    <xs:enumeration value="vertical"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="anchor" type="UI-COMPONENT-ANCHOR" use="optional" default="west">
    <xs:enumeration value="north"></xs:enumeration>
    <xs:enumeration value="northeast"></xs:enumeration>

```

```

    <xs:enumeration value="east"></xs:enumeration>
    <xs:enumeration value="southeast"></xs:enumeration>
    <xs:enumeration value="south"></xs:enumeration>
    <xs:enumeration value="southwest"></xs:enumeration>
    <xs:enumeration value="west"></xs:enumeration>
    <xs:enumeration value="northwest"></xs:enumeration>
    <xs:enumeration value="center"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="leftIndent" type="xs:nonNegativeInteger"
    use="optional"></xs:attribute>
</xs:element>

```

父 元素

[ActionButton 元素](#), [CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#),
[ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#),
[ComboBoxControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元
素](#), [ExtensionObjectPanel 元素](#), [ModelViewerPanel 元素](#), [MultiFieldChooserControl
元素](#), [MultiItemChooserControl 元素](#), [PasswordBoxControl 元素](#),
[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [PropertyControl 元素](#),
[RadioButtonGroupControl 元素](#), [SelectorPanel 元素](#), [ServerDirectoryChooserControl
元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#),
[SingleFieldValueChooserControl 元素](#), [SingleItemChooserControl 元素](#),
[SpinnerControl 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#),
[TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#), [TextBrowserPanel
元素](#), [ItemChooserControl 类型](#)

子 元素

[Cell 元素](#)

Cell 元素

表 A-72
属性 Cell

属性	使用	说明	有效值
column	必须的		nonNegativeInteger
row	必须的		nonNegativeInteger
width	必须的		nonNegativeInteger

XML 表示

```

<xs:element name="Cell">
  <xs:attribute name="row" type="xs:nonNegativeInteger" use="required"></xs:attribute>
  <xs:attribute name="column" type="xs:nonNegativeInteger"
    use="required"></xs:attribute>
  <xs:attribute name="width" type="xs:nonNegativeInteger"
    use="required"></xs:attribute>
</xs:element>

```


父 元素[Layout 元素](#)**License 元素**

Reserved for system use.

XML 表示

```

<xs:element name="License">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="OptionCode"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素[Extension 元素](#)**子 元素**[OptionCode 元素](#)**ListValue 元素**

A sequence of values. All values must have the same content type but this is not checked.

XML 表示

```

<xs:element name="ListValue" type="LIST-VALUE">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>

```

父 元素[Attribute 元素](#), [ListValue 元素](#), [Parameter 元素](#), [Attribute 元素](#), [ListValue 元素](#)**子 元素**[DatabaseConnectionValue 元素](#), [ListValue 元素](#), [MapValue 元素](#), [StructuredValue 元素](#), [Value 元素](#)

MapValue 元素

A set of map entries, each consisting if a key and a value.

XML 表示

```

<xs:element name="MapValue" type="MAP-VALUE">
  <xs:sequence>
    <xs:element name="MapEntry" type="MAP-ENTRY" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="KeyValue" type="KEY-VALUE"></xs:element>
        <xs:element name="StructuredValue" type="STRUCTURED-VALUE">
          <xs:sequence>
            <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
              <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
                <xs:choice>
                  <xs:element ref="MapValue"></xs:element>
                  <xs:element ref="StructuredValue"></xs:element>
                  <xs:element ref="ListValue"></xs:element>
                  <xs:element ref="Value"></xs:element>
                  <xs:element ref="DatabaseConnectionValue"></xs:element>
                </xs:choice>
              </xs:group>
            </xs:sequence>
            <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0"
              maxOccurs="1">
              <xs:group ref="PARAMETER-CONTENT" minOccurs="0"
                maxOccurs="unbounded">
                <xs:choice>
                  <xs:element ref="MapValue"></xs:element>
                  <xs:element ref="StructuredValue"></xs:element>
                  <xs:element ref="ListValue"></xs:element>
                  <xs:element ref="Value"></xs:element>
                  <xs:element ref="DatabaseConnectionValue"></xs:element>
                </xs:choice>
              </xs:group>
            </xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[ListValue 元素](#), [Attribute 元素](#), [ListValue 元素](#), [Parameter 元素](#), [Attribute 元素](#), [ListValue 元素](#)

子 元素

[MapEntry 元素](#)

MapEntry 元素

An entry in a keyed property map. Each entry consists of a key and an associated value.

XML 表示

```
<xs:element name="MapEntry" type="MAP-ENTRY" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="KeyValue" type="KEY-VALUE"></xs:element>
    <xs:element name="StructuredValue" type="STRUCTURED-VALUE">
      <xs:sequence>
        <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
          <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
            <xs:choice>
              <xs:element ref="MapValue"></xs:element>
              <xs:element ref="StructuredValue"></xs:element>
              <xs:element ref="ListValue"></xs:element>
              <xs:element ref="Value"></xs:element>
              <xs:element ref="DatabaseConnectionValue"></xs:element>
            </xs:choice>
          </xs:group>
        </xs:sequence>
        <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0"
          maxOccurs="1">
          <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
            <xs:choice>
              <xs:element ref="MapValue"></xs:element>
              <xs:element ref="StructuredValue"></xs:element>
              <xs:element ref="ListValue"></xs:element>
              <xs:element ref="Value"></xs:element>
              <xs:element ref="DatabaseConnectionValue"></xs:element>
            </xs:choice>
          </xs:group>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[MapValue 元素](#)

子 元素

[KeyValue 元素](#), [StructuredValue 元素](#)

KeyValue 元素

The key value in a map entry.

表 A-73
属性 KeyValue

属性	使用	说明	有效值
value	必须的		string

XML 表示

```
<xs:element name="KeyValue" type="KEY-VALUE">
  <xs:attribute name="value" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

父 元素

[MapEntry 元素](#)

StructuredValue 元素

A sequence of named values (“attributes”).

XML 表示

```
<xs:element name="StructuredValue" type="STRUCTURED-VALUE">
  <xs:sequence>
    <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"/></xs:element>
          <xs:element ref="StructuredValue"/></xs:element>
          <xs:element ref="ListValue"/></xs:element>
          <xs:element ref="Value"/></xs:element>
          <xs:element ref="DatabaseConnectionValue"/></xs:element>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/></xs:element>
          <xs:element ref="StructuredValue"/></xs:element>
          <xs:element ref="ListValue"/></xs:element>
          <xs:element ref="Value"/></xs:element>
          <xs:element ref="DatabaseConnectionValue"/></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[MapEntry 元素](#)

子 元素

Attribute 元素

Attribute 元素

表 A-74
属性 Attribute

属性	使用	说明	有效值
name	必须的		string
value	可选的		string

XML 表示

```

<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"></xs:element>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

StructuredValue 元素

子 元素

DatabaseConnectionValue 元素, ListValue 元素, ListValue 元素, MapValue 元素, StructuredValue 元素, Value 元素

ListValue 元素

A sequence of values. All values must have the same content type but this is not checked.

XML 表示

```
<xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

父 元素

[Attribute 元素](#)

子 元素

[DatabaseConnectionValue 元素](#), [ListValue 元素](#), [MapValue 元素](#), [StructuredValue 元素](#), [Value 元素](#)

Menu 元素

表 A-75
属性 Menu

属性	使用	说明	有效值
id	必须的		string
label	必须的		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string
offset	optional		nonNegativeInteger
separatorAfter	optional		boolean
separatorBefore	optional		boolean
showIcon	optional		boolean

属性	使用	说明	有效值
showLabel	optional		boolean
systemMenu	必须的		file edit insert view tools window help generate file.project file.outputs file.models edit.stream edit.node edit.outputs edit.models edit.project tools.repository tools.options tools.streamProperties

XML 表示

```

<xs:element name="Menu">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="required">
    <xs:enumeration value="file"></xs:enumeration>
    <xs:enumeration value="edit"></xs:enumeration>
    <xs:enumeration value="insert"></xs:enumeration>
    <xs:enumeration value="view"></xs:enumeration>
    <xs:enumeration value="tools"></xs:enumeration>
    <xs:enumeration value="window"></xs:enumeration>
    <xs:enumeration value="help"></xs:enumeration>
    <xs:enumeration value="generate"></xs:enumeration>
    <xs:enumeration value="file.project"></xs:enumeration>
    <xs:enumeration value="file.outputs"></xs:enumeration>
    <xs:enumeration value="file.models"></xs:enumeration>
    <xs:enumeration value="edit.stream"></xs:enumeration>
    <xs:enumeration value="edit.node"></xs:enumeration>
    <xs:enumeration value="edit.outputs"></xs:enumeration>
    <xs:enumeration value="edit.models"></xs:enumeration>
    <xs:enumeration value="edit.project"></xs:enumeration>
    <xs:enumeration value="tools.repository"></xs:enumeration>
    <xs:enumeration value="tools.options"></xs:enumeration>
    <xs:enumeration value="tools.streamProperties"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional"
    default="false"></xs:attribute>

```

```

<xs:attribute name="separatorBefore" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="separatorAfter" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional"
default="0"></xs:attribute>
</xs:element>

```

父 元素

[Controls 元素](#)

MenuItem 元素

表 A-76

属性 MenuItem

属性	使用	说明	有效值
action	必须的		string
customMenu	optional		string
offset	optional		nonNegativeInteger
separatorAfter	optional		boolean
separatorBefore	optional		boolean
showIcon	optional		boolean
showLabel	optional		boolean
systemMenu	optional		file edit insert view tools window help generate file.project file.outputs file.models edit.stream edit.node edit.outputs edit.models edit.project tools.repository tools.options tools.streamProperties

XML 表示

```

<xs:element name="MenuItem">
  <xs:attribute name="action" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="optional">
    <xs:enumeration value="file"></xs:enumeration>
    <xs:enumeration value="edit"></xs:enumeration>
    <xs:enumeration value="insert"></xs:enumeration>
    <xs:enumeration value="view"></xs:enumeration>
  </xs:attribute>
</xs:element>

```



```

    <xs:enumeration value="tools"></xs:enumeration>
    <xs:enumeration value="window"></xs:enumeration>
    <xs:enumeration value="help"></xs:enumeration>
    <xs:enumeration value="generate"></xs:enumeration>
    <xs:enumeration value="file.project"></xs:enumeration>
    <xs:enumeration value="file.outputs"></xs:enumeration>
    <xs:enumeration value="file.models"></xs:enumeration>
    <xs:enumeration value="edit.stream"></xs:enumeration>
    <xs:enumeration value="edit.node"></xs:enumeration>
    <xs:enumeration value="edit.outputs"></xs:enumeration>
    <xs:enumeration value="edit.models"></xs:enumeration>
    <xs:enumeration value="edit.project"></xs:enumeration>
    <xs:enumeration value="tools.repository"></xs:enumeration>
    <xs:enumeration value="tools.options"></xs:enumeration>
    <xs:enumeration value="tools.streamProperties"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="customMenu" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional"
  default="0"></xs:attribute>
</xs:element>

```

父 元素

[Controls 元素](#)

MissingValues 元素

表 A-77
属性 MissingValues

属性	使用	说明	有效值
treatNullAsMissing	可选的		boolean
treatWhitespaceAsMissing	可选的		boolean

XML 表示

```

<xs:element name="MissingValues" type="MISSING-VALUES" minOccurs="0">
  <xs:sequence>
    <xs:element name="Range" type="RANGE"></xs:element>
    <xs:element name="Values" type="FIELD-VALUES">
      <xs:sequence>
        <xs:element name="Value" type="FIELD-VALUE" minOccurs="0"
        maxOccurs="unbounded">
          <xs:sequence>
            <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
            maxOccurs="unbounded"></xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

```

        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean"
  default="true"></xs:attribute>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean"
  default="false"></xs:attribute>
</xs:element>

```

父 元素

[Field 元素](#)

子 元素

[Range 元素](#), [Values 元素](#)

Range 元素

表 A-78
属性 Range

属性	使用	说明	有效值
maxValue	必须的		string
minValue	必须的		string

XML 表示

```

<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="maxValue" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[MissingValues 元素](#)

Values 元素

表 A-79
属性 Values

属性	使用	说明	有效值
count	可选的		nonNegativeInteger

XML 表示

```

<xs:element name="Values" type="FIELD-VALUES">
  <xs:sequence>
    <xs:element name="Value" type="FIELD-VALUE" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
        maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

```

        </xs:sequence>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
  </xs:element>

```

父 元素

[MissingValues 元素](#)

子 元素

[Value 元素](#)

Value 元素

表 A-80
属性 Value

属性	使用	说明	有效值
code	必须的		integer
displayLabel	可选的		string
flagValue	可选的		boolean
value	必须的		string

XML 表示

```

<xs:element name="Value" type="FIELD-VALUE" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="flagValue" type="xs:boolean"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[Values 元素](#)

子 元素

[DisplayLabel 元素](#)

DisplayLabel 元素

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

表 A-81
属性 DisplayLabel

属性	使用	说明	有效值
lang	可选的		NMTOKEN
value	必须的		string

XML 表示

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>
```

父 元素

[Value 元素](#)

ModelBuilder 元素

表 A-82
属性 ModelBuilder

属性	使用	说明	有效值
allowNoInputs	optional		boolean
allowNoOutputs	optional		boolean
miningFunctions	必须的		任意
nullifyBlanks	optional		boolean

XML 表示

```
<xs:element name="ModelBuilder">
  <xs:sequence>
    <xs:element name="Algorithm"></xs:element>
    <xs:element name="ModelingFields" minOccurs="0">
      <xs:sequence>
        <xs:element name="InputFields" minOccurs="0"></xs:element>
        <xs:element name="OutputFields" minOccurs="0"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="ModelGeneration"></xs:element>
    <xs:element name="ModelFields">
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="DATA-MODEL-EXPRESSION">
          <xs:choice>
            <xs:element ref="ForEach"></xs:element>
            <xs:element ref="AddField"></xs:element>
            <xs:element ref="ChangeField"></xs:element>
            <xs:element ref="RemoveField"></xs:element>
          </xs:choice>
        </xs:group>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
```

```

<xs:element name="ModelEvaluation" minOccurs="0">
  <xs:sequence>
    <xs:element name="RawPropensity" minOccurs="0"></xs:element>
    <xs:element name="AdjustedPropensity" minOccurs="0"></xs:element>
    <xs:element name="VariableImportance" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:element>
<xs:element name="AutoModeling" minOccurs="0">
  <xs:sequence>
    <xs:element name="SimpleSettings">
      <xs:sequence>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="ExpertSettings" minOccurs="0">
      <xs:sequence>
        <xs:element ref="Condition" minOccurs="0"></xs:element>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="PropertyMap" minOccurs="0">
      <xs:sequence>
        <xs:element name="PropertyMapping" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element ref="Constraint" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
</xs:sequence>
<xs:attribute name="miningFunctions" use="required"></xs:attribute>
<xs:attribute name="allowNoInputs" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
<xs:attribute name="allowNoOutputs" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
<xs:attribute name="nullifyBlanks" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
</xs:element>

```

父 元素

[Node 元素](#)

子 元素

[Algorithm 元素](#), [AutoModeling 元素](#), [ModelEvaluation 元素](#), [ModelFields 元素](#),
[ModelGeneration 元素](#), [ModelingFields 元素](#)

Algorithm 元素

表 A-83
属性 Algorithm

属性	使用	说明	有效值
label	必须的		string
labelKey	optional		string
value	必须的		string

XML 表示

```
<xs:element name="Algorithm">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[ModelBuilder 元素](#)

ModelingFields 元素

表 A-84
属性 ModelingFields

属性	使用	说明	有效值
controlsId	必须的		string
ignoreBOTH	optional		boolean

XML 表示

```
<xs:element name="ModelingFields" minOccurs="0">
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
  <xs:sequence>
    <xs:element name="InputFields" minOccurs="0"></xs:element>
    <xs:element name="OutputFields" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="ignoreBOTH" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
</xs:element>
```

父 元素

[ModelBuilder 元素](#)

子 元素

[InputFields 元素](#), [OutputFields 元素](#)

相关 元素

[DocumentGeneration 元素](#), [ModelGeneration 元素](#), [ModelEvaluation 元素](#)

InputFields 元素

表 A-85
属性 InputFields

属性	使用	说明	有效值
label	必须的		string
labelKey	optional		string
multiple	必须的		boolean
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	必须的		string
storage	optional		string
types	optional		string

XML 表示

```
<xs:element name="InputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="multiple" type="xs:boolean" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[ModelingFields 元素](#)

相关 元素

[OutputFields 元素](#)

OutputFields 元素

表 A-86
属性 OutputFields

属性	使用	说明	有效值
label	必须的		string
labelKey	optional		string
multiple	必须的		boolean
onlyDatetime	optional		boolean

属性	使用	说明	有效值
onlyDiscrete	optional		boolean
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	必须的		string
storage	optional		string
types	optional		string

XML 表示

```

<xs:element name="OutputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="multiple" type="xs:boolean" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[ModelingFields 元素](#)

相关 元素

[InputFields 元素](#)

ModelGeneration 元素

表 A-87

属性 ModelGeneration

属性	使用	说明	有效值
controlsId	必须的		string

XML 表示

```

<xs:element name="ModelGeneration">
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[ModelBuilder 元素](#)

相关 元素

[DocumentGeneration](#) 元素, [ModelingFields](#) 元素, [ModelEvaluation](#) 元素

ModelFields 元素

XML 表示

```
<xs:element name="ModelFields">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"></xs:element>
        <xs:element ref="AddField"></xs:element>
        <xs:element ref="ChangeField"></xs:element>
        <xs:element ref="RemoveField"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

父 元素

[ModelBuilder](#) 元素

子 元素

[AddField](#) 元素, [ChangeField](#) 元素, [ForEach](#) 元素, [RemoveField](#) 元素

ModelEvaluation 元素

表 A-88
属性 ModelEvaluation

属性	使用	说明	有效值
container	必须的		任意
controlsId	必须的		string
outputContainer	optional		任意

XML 表示

```
<xs:element name="ModelEvaluation" minOccurs="0">
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
  <xs:sequence>
    <xs:element name="RawPropensity" minOccurs="0"></xs:element>
    <xs:element name="AdjustedPropensity" minOccurs="0"></xs:element>
    <xs:element name="VariableImportance" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="container" use="required"></xs:attribute>
  <xs:attribute name="outputContainer" use="optional"></xs:attribute>
</xs:element>
```

父 元素[ModelBuilder 元素](#)**子 元素**[AdjustedPropensity 元素](#), [RawPropensity 元素](#), [VariableImportance 元素](#)**相关 元素**[DocumentGeneration 元素](#), [ModelingFields 元素](#), [ModelGeneration 元素](#)**RawPropensity 元素**

表 A-89

属性 RawPropensity

属性	使用	说明	有效值
availabilityProperty	optional		string
defaultValue	optional		boolean
property	optional		string

XML 表示

```
<xs:element name="RawPropensity" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string"
    use="optional"></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>
```

父 元素[ModelEvaluation 元素](#)**AdjustedPropensity 元素**

表 A-90

属性 AdjustedPropensity

属性	使用	说明	有效值
availabilityProperty	optional		string
defaultValue	optional		boolean
property	optional		string

XML 表示

```
<xs:element name="AdjustedPropensity" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string"
    use="optional"></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>
```

父 元素[ModelEvaluation 元素](#)**VariableImportance 元素**

表 A-91

属性 VariableImportance

属性	使用	说明	有效值
availabilityProperty	optional		string
defaultValue	optional		boolean
property	optional		string

XML 表示

```
<xs:element name="VariableImportance" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string"
    use="optional"></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>
```

父 元素[ModelEvaluation 元素](#)**AutoModeling 元素**

表 A-92

属性 AutoModeling

属性	使用	说明	有效值
enabledByDefault	optional		任意

XML 表示

```
<xs:element name="AutoModeling" minOccurs="0">
  <xs:sequence>
    <xs:element name="SimpleSettings">
      <xs:sequence>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="ExpertSettings" minOccurs="0">
      <xs:sequence>
        <xs:element ref="Condition" minOccurs="0"></xs:element>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="PropertyMap" minOccurs="0">
      <xs:sequence>
        <xs:element name="PropertyMapping" maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

```
    <xs:element ref="Constraint" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="enabledByDefault" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[ModelBuilder 元素](#)

子 元素

[Constraint 元素](#), [ExpertSettings 元素](#), [PropertyMap 元素](#), [SimpleSettings 元素](#)

SimpleSettings 元素

XML 表示

```
<xs:element name="SimpleSettings">
  <xs:sequence>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[AutoModeling 元素](#)

子 元素

[PropertyGroup 元素](#)

ExpertSettings 元素

XML 表示

```
<xs:element name="ExpertSettings" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Condition" minOccurs="0"></xs:element>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[AutoModeling 元素](#)

子 元素

[Condition 元素](#), [PropertyGroup 元素](#)

PropertyMap 元素

XML 表示

```
<xs:element name="PropertyMap" minOccurs="0">
  <xs:sequence>
    <xs:element name="PropertyMapping" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[AutoModeling 元素](#)

子 元素

[PropertyMapping 元素](#)

PropertyMapping 元素

表 A-93

属性 PropertyMapping

属性	使用	说明	有效值
property	必须的		string
systemProperty	必须的		string

XML 表示

```
<xs:element name="PropertyMapping" maxOccurs="unbounded">
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="systemProperty" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[PropertyMap 元素](#)

ModelOutput 元素

表 A-94

属性 ModelOutput

属性	使用	说明	有效值
deprecatedScriptNames	optional		string
helpLink	optional		string
id	必须的		string
label	optional		string
labelKey	optional		string
scriptName	optional		string

XML 表示

```

<xs:element name="ModelOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string"
  use="optional"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Extension 元素](#)

子 元素

[Constructors 元素](#), [Containers 元素](#), [ModelProvider 元素](#), [Properties 元素](#), [UserInterface 元素](#)

相关 元素

[DocumentOutput 元素](#), [InteractiveDocumentBuilder 元素](#), [InteractiveModelBuilder 元素](#), [Node 元素](#)

Containers 元素**XML 表示**

```

<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[ModelOutput 元素](#)

子 元素

[Container](#) 元素

ModelProvider 元素

表 A-95

属性 ModelProvider

属性	使用	说明	有效值
container	必须的		string
isPMML	optional		boolean

XML 表示

```
<xs:element name="ModelProvider">
  <xs:attribute name="container" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="isPMML" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
</xs:element>
```

父 元素

[DocumentOutput](#) 元素, [InteractiveDocumentBuilder](#) 元素, [InteractiveModelBuilder](#) 元素, [ModelOutput](#) 元素, [Node](#) 元素

ModelType 元素

Defines a new model type

表 A-96

属性 ModelType

属性	使用	说明	有效值
format	必须的		utf8 binary
id	必须的		string
inputDirection	optional		string
outputDirection	optional		string

XML 表示

```
<xs:element name="ModelType">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"></xs:enumeration>
        <xs:enumeration value="binary"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
```

```

    <xs:attribute name="inputDirection" type="xs:string" use="optional"
    default="[in]"></xs:attribute>
    <xs:attribute name="outputDirection" type="xs:string" use="optional"
    default="[out]"></xs:attribute>
</xs:element>

```

父 元素

[ContainerTypes 元素](#)

相关 元素

[DocumentType 元素](#)

ModelViewerPanel 元素

表 A-97
属性 ModelViewerPanel

属性	使用	说明	有效值
container	必须的		string

XML 表示

```

<xs:element name="ModelViewerPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[Tab 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[SelectorPanel 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#),
[TextBrowserPanel 元素](#)

Module 元素

表 A-98
属性 Module

属性	使用	说明	有效值
libraryId	optional		string
name	optional		string
systemModule	optional		SmartScore

XML 表示

```
<xs:element name="Module">
  <xs:sequence>
    <xs:element ref="InputFiles"></xs:element>
    <xs:element ref="OutputFiles"></xs:element>
    <xs:element ref="StatusCodes" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:sequence>
  <xs:attribute name="systemModule" use="optional" default="SmartScore">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="SmartScore"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="libraryId" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="name" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Execution 元素](#)

子 元素

[InputFiles 元素](#), [OutputFiles 元素](#), [StatusCodes 元素](#)

MultiFieldChooserControl 元素

表 A-99
属性 MultiFieldChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean

属性	使用	说明	有效值
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	必须的		string
showLabel	optional		boolean
storage	optional		string
types	optional		string

XML 表示

```

<xs:element name="MultiFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="storage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl](#) 元素, [CheckBoxGroupControl](#) 元素, [ClientDirectoryChooserControl](#) 元素, [ClientFileChooserControl](#) 元素, [DBConnectionChooserControl](#) 元素, [DBTableChooserControl](#) 元素, [PasswordBoxControl](#) 元素, [PropertyControl](#) 元素, [RadioButtonGroupControl](#) 元素, [ServerDirectoryChooserControl](#) 元素, [ServerFileChooserControl](#) 元素, [SingleFieldChooserControl](#) 元素, [SingleFieldValueChooserControl](#) 元素, [SpinnerControl](#) 元素, [TableControl](#) 元素, [TextAreaControl](#) 元素, [TextBoxControl](#) 元素

MultiltemChooserControl 元素

表 A-100

属性 MultiltemChooserControl

属性	使用	说明	有效值
catalog	必须的		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="MultiltemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="catalog" type="xs:string" use="required"></xs:attribute>

```

```
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[SingleItemChooserControl 元素](#)

Node 元素

表 A-101
属性 Node

属性	使用	说明	有效值
deprecatedScriptNames	optional		string
description	optional		string
descriptionKey	optional		string
helpLink	optional		string
id	必须的		string
label	必须的		string
labelKey	optional		string
palette	optional		import fieldOp recordOp modeling dbModeling graph output export modeling.classification modeling.association modeling.segmentation modeling.auto
relativePosition	optional		string
relativeTo	optional		string
scriptName	optional		string
type	必须的		dataReader dataWriter dataTransformer modelApplier modelBuilder documentBuilder

XML 表示

```

<xs:element name="Node">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string"
  use="optional"></xs:attribute>
  <xs:sequence>
    <xs:element ref="ModelBuilder" minOccurs="0"></xs:element>
    <xs:element ref="DocumentBuilder" minOccurs="0"></xs:element>
    <xs:element ref="Execution"></xs:element>
    <xs:element ref="OutputDataModel" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="type" type="NODE-TYPE" use="required">
    <xs:enumeration value="dataReader"></xs:enumeration>
    <xs:enumeration value="dataWriter"></xs:enumeration>
    <xs:enumeration value="dataTransformer"></xs:enumeration>
    <xs:enumeration value="modelApplier"></xs:enumeration>
    <xs:enumeration value="modelBuilder"></xs:enumeration>
    <xs:enumeration value="documentBuilder"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="palette" type="SYSTEM-PALETTE" use="optional">
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="fieldOp"></xs:enumeration>
    <xs:enumeration value="recordOp"></xs:enumeration>
    <xs:enumeration value="modeling"></xs:enumeration>
    <xs:enumeration value="dbModeling"></xs:enumeration>
    <xs:enumeration value="graph"></xs:enumeration>
    <xs:enumeration value="output"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
    <xs:enumeration value="modeling.classification"></xs:enumeration>
    <xs:enumeration value="modeling.association"></xs:enumeration>
    <xs:enumeration value="modeling.segmentation"></xs:enumeration>
    <xs:enumeration value="modeling.auto"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="relativeTo" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="relativePosition" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Extension 元素](#)

子 元素

[Constructors 元素](#), [Containers 元素](#), [DocumentBuilder 元素](#), [Execution 元素](#), [ModelBuilder 元素](#), [ModelProvider 元素](#), [OutputDataModel 元素](#), [Properties 元素](#), [UserInterface 元素](#)

相关 元素

[DocumentOutput 元素](#), [InteractiveDocumentBuilder 元素](#), [InteractiveModelBuilder 元素](#), [ModelOutput 元素](#)

Containers 元素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[Node 元素](#)

子 元素

[Container 元素](#)

Not 元素

XML 表示

```
<xs:element name="Not">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

父 元素

And 元素, Command 元素, Constraint 元素, CreateDocument 元素, CreateDocumentOutput 元素, CreateInteractiveDocumentBuilder 元素, CreateInteractiveModelBuilder 元素, CreateModel 元素, CreateModelApplier 元素, CreateModelOutput 元素, Enabled 元素, Option 元素, Or 元素, Run 元素, Visible 元素

子 元素

And 元素, Condition 元素, Not 元素, Or 元素

NumberFormat 元素

Defines format information for a numeric field.

表 A-102

属性 NumberFormat

属性	使用	说明	有效值
decimalPlaces	必须的		nonNegativeInteger
decimalSymbol	必须的		period comma
formatType	必须的		standard scientific currency
groupingSymbol	必须的		none period comma space
name	必须的		string

XML 表示

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger"
    use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

NumericInfo 元素

表 A-103
属性 NumericInfo

属性	使用	说明	有效值
mean	可选的		double
standardDeviation	可选的		double

XML 表示

```
<xs:element name="NumericInfo">
  <xs:attribute name="mean" type="xs:double"></xs:attribute>
  <xs:attribute name="standardDeviation" type="xs:double"></xs:attribute>
</xs:element>
```

父 元素

[AddField 元素](#), [ChangeField 元素](#), [Field 元素](#)

Option 元素

表 A-104
属性 Option

属性	使用	说明	有效值
ifProperty	optional		string
unlessProperty	optional		string
value	必须的		

XML 表示

```
<xs:element name="Option">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="value" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:attribute name="ifProperty" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="unlessProperty" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Run 元素](#)

子 元素

[And 元素](#), [Condition 元素](#), [Not 元素](#), [Or 元素](#)

OptionCode 元素

表 A-105

属性 OptionCode

属性	使用	说明	有效值
code	可选的		long
description	可选的		string
type	可选的		mandatory optional

XML 表示

```

<xs:element name="OptionCode">
  <xs:attribute name="code" type="xs:long"></xs:attribute>
  <xs:attribute name="type" type="LicenseType">
    <xs:enumeration value="mandatory"></xs:enumeration>
    <xs:enumeration value="optional"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="description" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[License 元素](#)

Or 元素

XML 表示

```

<xs:element name="Or">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

父 元素

[And 元素](#), [Command 元素](#), [Constraint 元素](#), [CreateDocument 元素](#),
[CreateDocumentOutput 元素](#), [CreateInteractiveDocumentBuilder 元素](#),
[CreateInteractiveModelBuilder 元素](#), [CreateModel 元素](#), [CreateModelApplier 元](#)

素, [CreateModelOutput](#) 元素, [Enabled](#) 元素, [Not](#) 元素, [Option](#) 元素, [Run](#) 元素, [Visible](#) 元素

子 元素

[And](#) 元素, [Condition](#) 元素, [Not](#) 元素, [Or](#) 元素

OutputDataModel 元素

表 A-106
属性 OutputDataModel

属性	使用	说明	有效值
libraryId	optional		string
method	optional		xml dataModelProvider sharedLibrary
mode	optional		fixed modify extend replace
providerClass	optional		string

XML 表示

```
<xs:element name="OutputDataModel">
  <xs:attribute name="mode" use="optional" default="fixed">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="fixed"></xs:enumeration>
        <xs:enumeration value="modify"></xs:enumeration>
        <xs:enumeration value="extend"></xs:enumeration>
        <xs:enumeration value="replace"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="method" use="optional" default="xml">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="xml"></xs:enumeration>
        <xs:enumeration value="dataModelProvider"></xs:enumeration>
        <xs:enumeration value="sharedLibrary"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="providerClass" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="libraryId" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Node](#) 元素

OutputFiles 元素

XML 表示

```
<xs:element name="OutputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"></xs:element>
      <xs:element ref="ContainerFile" minOccurs="0"
        maxOccurs="unbounded"></xs:element>
    </xs:sequence>
  </xs:group>
</xs:element>
```

父 元素

[Execution 元素](#), [Module 元素](#)

子 元素

[ContainerFile 元素](#), [DataFile 元素](#)

Palette 元素

表 A-107
属性 Palette

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
id	必须的		string
label	必须的		string
labelKey	optional		string
position	optional		atStart atEnd before after
systemPalette	optional		import fieldOp recordOp modeling dbModeling graph output export modeling.classification modeling.association modeling.segmentation modeling.auto

XML 表示

```

<xs:element name="Palette">
  <xs:sequence>
    <xs:element ref="Icon"></xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="position" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="atStart"></xs:enumeration>
        <xs:enumeration value="atEnd"></xs:enumeration>
        <xs:enumeration value="before"></xs:enumeration>
        <xs:enumeration value="after"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="systemPalette" type="SYSTEM-PALETTE" use="optional">
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="fieldOp"></xs:enumeration>
    <xs:enumeration value="recordOp"></xs:enumeration>
    <xs:enumeration value="modeling"></xs:enumeration>
    <xs:enumeration value="dbModeling"></xs:enumeration>
    <xs:enumeration value="graph"></xs:enumeration>
    <xs:enumeration value="output"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
    <xs:enumeration value="modeling.classification"></xs:enumeration>
    <xs:enumeration value="modeling.association"></xs:enumeration>
    <xs:enumeration value="modeling.segmentation"></xs:enumeration>
    <xs:enumeration value="modeling.auto"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

子 元素

[Icon 元素](#)

Parameters 元素

Configuration parameters from the extension node.

表 A-108

属性 Parameters

属性	使用	说明	有效值
count	可选的		nonNegativeInteger

XML 表示

```

<xs:element name="Parameters" type="PARAMETERS">
  <xs:sequence>
    <xs:element name="Parameter" type="PARAMETER" minOccurs="0" maxOccurs="unbounded">

```

```

    <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
      <xs:choice>
        <xs:element ref="MapValue"></xs:element>
        <xs:element ref="StructuredValue"></xs:element>
        <xs:element ref="ListValue"></xs:element>
        <xs:element ref="Value"></xs:element>
        <xs:element ref="DatabaseConnectionValue"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:element>
</xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

子 元素

[Parameter 元素](#)

Parameter 元素

A parameter has a name and a value. A simple value can be expressed with the value attribute; a compound value uses the content model described by ParameterContent. This combination of attribute and content is repeated for nested values.

表 A-109
属性 Parameter

属性	使用	说明	有效值
name	必须的		string
value	可选的		string

XML 表示

```

<xs:element name="Parameter" type="PARAMETER" minOccurs="0" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[Parameters 元素](#)

子 元素

[DatabaseConnectionValue](#) 元素, [ListValue](#) 元素, [MapValue](#) 元素, [StructuredValue](#) 元素, [Value](#) 元素

PasswordBoxControl 元素

表 A-110

属性 PasswordBoxControl

属性	使用	说明	有效值
columns	optional		positiveInteger
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="PasswordBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional"
  default="20"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel](#) 元素, [PropertiesSubPanel](#) 元素

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

Properties 元素**XML 表示**

```
<xs:element name="Properties">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[DocumentOutput 元素](#), [Execution 元素](#), [InteractiveDocumentBuilder 元素](#), [InteractiveModelBuilder 元素](#), [ModelOutput 元素](#), [Node 元素](#)

子 元素

[Property 元素](#)

PropertiesPanel 元素

表 A-111
属性 PropertiesPanel

属性	使用	说明	有效值
id	optional		string
label	optional		string
labelKey	optional		string

XML 表示

```
<xs:element name="PropertiesPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

</xs:sequence>
<xs:sequence maxOccurs="unbounded">
  <xs:choice>
    <xs:element ref="CheckBoxControl"></xs:element>
    <xs:element ref="TextBoxControl"></xs:element>
    <xs:element ref="PasswordBoxControl"></xs:element>
    <xs:element ref="TextAreaControl"></xs:element>
    <xs:element ref="RadioButtonGroupControl"></xs:element>
    <xs:element ref="CheckBoxGroupControl"></xs:element>
    <xs:element ref="ComboBoxControl"></xs:element>
    <xs:element ref="SpinnerControl"></xs:element>
    <xs:element ref="ServerFileChooserControl"></xs:element>
    <xs:element ref="ServerDirectoryChooserControl"></xs:element>
    <xs:element ref="ClientFileChooserControl"></xs:element>
    <xs:element ref="ClientDirectoryChooserControl"></xs:element>
    <xs:element ref="TableControl"></xs:element>
    <xs:element ref="SingleFieldChooserControl"></xs:element>
    <xs:element ref="MultiFieldChooserControl"></xs:element>
    <xs:element ref="SingleFieldValueChooserControl"></xs:element>
    <xs:element ref="SingleItemChooserControl"></xs:element>
    <xs:element ref="MultiItemChooserControl"></xs:element>
    <xs:element ref="DBConnectionChooserControl"></xs:element>
    <xs:element ref="DBTableChooserControl"></xs:element>
    <xs:element ref="PropertyControl"></xs:element>
    <xs:element ref="StaticText"></xs:element>
    <xs:element ref="SystemControls"></xs:element>
    <xs:element ref="ActionButton"></xs:element>
    <xs:element ref="PropertiesPanel"></xs:element>
    <xs:element ref="PropertiesSubPanel"></xs:element>
    <xs:element ref="SelectorPanel"></xs:element>
    <xs:element ref="ExtensionObjectPanel"></xs:element>
  </xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesSubPanel 元素](#), [Tab 元素](#)

子 元素

[ActionButton 元素](#), [CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [ComboBoxControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [Enabled 元素](#), [ExtensionObjectPanel 元素](#), [Layout 元素](#), [MultiFieldChooserControl 元素](#), [MultiItemChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [SelectorPanel 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SingleItemChooserControl 元素](#),

[SpinnerControl 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#), [Visible 元素](#)

相关 元素

[PropertiesSubPanel 元素](#)

PropertiesSubPanel 元素

表 A-112

属性 PropertiesSubPanel

属性	使用	说明	有效值
buttonLabel	optional		string
buttonLabelKey	optional		string
dialogTitle	optional		string
dialogTitleKey	optional		string
helpLink	optional		string
mnemonic	optional		string
mnemonicKey	optional		string

XML 表示

```
<xs:element name="PropertiesSubPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CheckBoxControl"></xs:element>
      <xs:element ref="TextBoxControl"></xs:element>
      <xs:element ref="PasswordBoxControl"></xs:element>
      <xs:element ref="TextAreaControl"></xs:element>
      <xs:element ref="RadioButtonGroupControl"></xs:element>
      <xs:element ref="CheckBoxGroupControl"></xs:element>
      <xs:element ref="ComboBoxControl"></xs:element>
      <xs:element ref="SpinnerControl"></xs:element>
      <xs:element ref="ServerFileChooserControl"></xs:element>
      <xs:element ref="ServerDirectoryChooserControl"></xs:element>
      <xs:element ref="ClientFileChooserControl"></xs:element>
      <xs:element ref="ClientDirectoryChooserControl"></xs:element>
      <xs:element ref="TableControl"></xs:element>
      <xs:element ref="SingleFieldChooserControl"></xs:element>
      <xs:element ref="MultiFieldChooserControl"></xs:element>
      <xs:element ref="SingleFieldValueChooserControl"></xs:element>
      <xs:element ref="SingleItemChooserControl"></xs:element>
      <xs:element ref="MultiItemChooserControl"></xs:element>
      <xs:element ref="DBConnectionChooserControl"></xs:element>
      <xs:element ref="DBTableChooserControl"></xs:element>
      <xs:element ref="PropertyControl"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

    <xs:element ref="StaticText"></xs:element>
    <xs:element ref="SystemControls"></xs:element>
    <xs:element ref="ActionButton"></xs:element>
    <xs:element ref="PropertiesPanel"></xs:element>
    <xs:element ref="PropertiesSubPanel"></xs:element>
    <xs:element ref="SelectorPanel"></xs:element>
    <xs:element ref="ExtensionObjectPanel"></xs:element>
  </xs:choice>
</xs:sequence>
<xs:attribute name="buttonLabel" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="buttonLabelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="dialogTitle" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="dialogTitleKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#)

子 元素

[ActionButton 元素](#), [CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [ComboBoxControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [Enabled 元素](#), [ExtensionObjectPanel 元素](#), [Layout 元素](#), [MultiFieldChooserControl 元素](#), [MultiItemChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [SelectorPanel 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SingleItemChooserControl 元素](#), [SpinnerControl 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#), [Visible 元素](#)

相关 元素

[PropertiesPanel 元素](#)

Property 元素

表 A-113
属性 Property

属性	使用	说明	有效值
defaultValue	optional		
defaultValueKey	optional		string
deprecatedScriptNames	optional		string
description	optional		string
descriptionKey	optional		string
isList	optional		boolean

属性	使用	说明	有效值
label	optional		string
labelKey	optional		string
max	optional		string
min	optional		string
name	必须的		string
scriptName	optional		string
type	optional		string
valueType	可选的		string encryptedString fieldName integer double boolean date enum structure databaseConnection

XML 表示

```

<xs:element name="Property">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"></xs:element>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="encryptedString"></xs:enumeration>
    <xs:enumeration value="fieldName"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="double"></xs:enumeration>
    <xs:enumeration value="boolean"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="enum"></xs:enumeration>
    <xs:enumeration value="structure"></xs:enumeration>
    <xs:enumeration value="databaseConnection"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="min" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="max" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string"
    use="optional"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING"
    use="optional"></xs:attribute>
  <xs:attribute name="defaultValueKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Properties](#) 元素, [PropertySets](#) 元素

子 元素

[DefaultValue](#) 元素

相关 元素

[PropertyType](#) 元素

PropertyControl 元素

表 A-114

属性 PropertyControl

属性	使用	说明	有效值
controlClass	必须的		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="PropertyControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

```
<xs:attribute name="controlClass" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

PropertyGroup 元素

表 A-115

属性 PropertyGroup

属性	使用	说明	有效值
label	optional		string
labelKey	optional		string
properties	必须的		string

XML 表示

```
<xs:element name="PropertyGroup">
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="properties" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[SimpleSettings 元素](#), [ExpertSettings 元素](#)

PropertySets 元素

XML 表示

```
<xs:element name="PropertySets">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
```

```
</xs:element>
```

父 元素

[CommonObjects](#) 元素

子 元素

[Property](#) 元素

PropertyType 元素

表 A-116
属性 PropertyType

属性	使用	说明	有效值
id	必须的		string
isKeyed	optional		boolean
isList	optional		boolean
max	optional		string
min	optional		string
valueType	可选的		string encryptedString fieldName integer double boolean date enum structure databaseConnection

XML 表示

```
<xs:element name="PropertyType">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"/></xs:element>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="encryptedString"/></xs:enumeration>
    <xs:enumeration value="fieldName"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="double"/></xs:enumeration>
    <xs:enumeration value="boolean"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="enum"/></xs:enumeration>
    <xs:enumeration value="structure"/></xs:enumeration>
    <xs:enumeration value="databaseConnection"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional"
    default="false"/></xs:attribute>
  <xs:attribute name="min" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="max" type="xs:string" use="optional"/></xs:attribute>
</xs:choice>
```

```

    <xs:element ref="Enumeration" minOccurs="0"></xs:element>
    <xs:element ref="Structure" minOccurs="0"></xs:element>
  </xs:choice>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="isKeyed" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
</xs:element>

```

父 元素

[PropertyTypes 元素](#)

子 元素

[DefaultValue 元素](#), [Enumeration 元素](#), [Structure 元素](#)

相关 元素

[Property 元素](#)

PropertyTypes 元素

XML 表示

```

<xs:element name="PropertyTypes">
  <xs:sequence>
    <xs:element ref="PropertyType" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[CommonObjects 元素](#)

子 元素

[PropertyType 元素](#)

RadioButtonGroupControl 元素

表 A-117
属性 RadioButtonGroupControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
falseLabel	optional		string
falseLabelKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger

属性	使用	说明	有效值
layoutByRow	optional		boolean
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
rows	optional		positiveInteger
showLabel	optional		boolean
trueFirst	optional		boolean
trueLabel	optional		string
trueLabelKey	optional		string
useSubPanel	optional		boolean

XML 表示

```

<xs:element name="RadioButtonGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="falseLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="falseLabelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueLabelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueFirst" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel](#) 元素, [PropertiesSubPanel](#) 元素

子 元素

[Enabled](#) 元素, [Layout](#) 元素, [Visible](#) 元素

相关 元素

[CheckBoxControl](#) 元素, [CheckBoxGroupControl](#) 元素, [ClientDirectoryChooserControl](#) 元素, [ClientFileChooserControl](#) 元素, [DBConnectionChooserControl](#) 元素, [DBTableChooserControl](#) 元素, [MultiFieldChooserControl](#) 元素, [PasswordBoxControl](#) 元素, [PropertyControl](#) 元素, [ServerDirectoryChooserControl](#) 元素, [ServerFileChooserControl](#) 元素, [SingleFieldChooserControl](#) 元素, [SingleFieldValueChooserControl](#) 元素, [SpinnerControl](#) 元素, [TableControl](#) 元素, [TextAreaControl](#) 元素, [TextBoxControl](#) 元素

Range 元素

表 A-118
属性 Range

属性	使用	说明	有效值
max	可选的		string
min	可选的		string

XML 表示

```
<xs:element name="Range">
  <xs:attribute name="min" type="xs:string"/></xs:attribute>
  <xs:attribute name="max" type="xs:string"/></xs:attribute>
</xs:element>
```

父 元素

[AddField](#) 元素, [MissingValues](#) 元素, [ChangeField](#) 元素, [MissingValues](#) 元素, [Field](#) 元素, [MissingValues](#) 元素

Range 元素

表 A-119
属性 Range

属性	使用	说明	有效值
maxValue	必须的		string
minValue	必须的		string

XML 表示

```
<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="maxValue" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

父 元素[Field 元素](#)**RemoveField 元素**

表 A-120

属性 RemoveField

属性	使用	说明	有效值
fieldRef	必须的		

XML 表示

```
<xs:element name="RemoveField">
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"/></xs:attribute>
</xs:element>
```

父 元素[ForEach 元素](#), [ModelFields 元素](#)**Resources 元素**

Defines common resources such as client-side libraries and resource bundles, and server-side libraries.

XML 表示

```
<xs:element name="Resources">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element name="Bundle" minOccurs="0"/></xs:element>
      <xs:element name="JarFile" minOccurs="0"/></xs:element>
      <xs:element name="SharedLibrary" minOccurs="0"/></xs:element>
      <xs:element name="HelpInfo" minOccurs="0"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

父 元素[Extension 元素](#)**子 元素**[Bundle 元素](#), [HelpInfo 元素](#), [JarFile 元素](#), [SharedLibrary 元素](#)

Bundle 元素

表 A-121
属性 Bundle

属性	使用	说明	有效值
id	必须的		string
path	必须的		
type	必须的		list properties

XML 表示

```
<xs:element name="Bundle" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="list"></xs:enumeration>
        <xs:enumeration value="properties"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

父 元素

[Resources 元素](#)

JarFile 元素

表 A-122
属性 JarFile

属性	使用	说明	有效值
id	必须的		string
path	必须的		

XML 表示

```
<xs:element name="JarFile" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

父 元素

[Resources 元素](#)

SharedLibrary 元素

表 A-123
属性 SharedLibrary

属性	使用	说明	有效值
id	必须的		string
path	必须的		

XML 表示

```
<xs:element name="SharedLibrary" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

父 元素

[Resources](#) 元素

HelpInfo 元素

表 A-124
属性 HelpInfo

属性	使用	说明	有效值
default	optional		string
helpset	optional		
id	optional		string
missing	optional		string
path	optional		
type	必须的		native javahelp html

XML 表示

```
<xs:element name="HelpInfo" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="native"></xs:enumeration>
        <xs:enumeration value="javahelp"></xs:enumeration>
        <xs:enumeration value="html"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="helpset" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="default" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="missing" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[Resources 元素](#)

Run 元素

XML 表示

```

<xs:element name="Run">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
    <xs:element ref="Command" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="Option" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="StatusCodes" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:element>

```

父 元素

[Executable 元素](#)

子 元素

[And 元素](#), [Command 元素](#), [Condition 元素](#), [Not 元素](#), [Option 元素](#), [Or 元素](#), [StatusCodes 元素](#)

SelectorPanel 元素

表 A-125

属性 SelectorPanel

属性	使用	说明	有效值
control	必须的		string

XML 表示

```

<xs:element name="SelectorPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Selector"></xs:element>
  </xs:sequence>

```

```
<xs:attribute name="control" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Selector 元素](#), [Visible 元素](#)

相关 元素

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[ModelViewerPanel 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#),
[TextBrowserPanel 元素](#)

Selector 元素

表 A-126
 属性 Selector

属性	使用	说明	有效值
controlValue	必须的		string
panelId	必须的		string

XML 表示

```
<xs:element name="Selector">
  <xs:attribute name="panelId" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="controlValue" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[SelectorPanel 元素](#)

ServerDirectoryChooserControl 元素

表 A-127
 属性 ServerDirectoryChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string

属性	使用	说明	有效值
mode	必须的		open save import export
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:element name="ServerDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/></xs:enumeration>
    <xs:enumeration value="save"/></xs:enumeration>
    <xs:enumeration value="import"/></xs:enumeration>
    <xs:enumeration value="export"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#),

SingleFieldValueChooserControl 元素, SpinnerControl 元素, TableControl 元素, TextAreaControl 元素, TextBoxControl 元素

ServerFileChooserControl 元素

表 A-128

属性 ServerFileChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
mode	必须的		open save import export
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="ServerFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"></xs:enumeration>
    <xs:enumeration value="save"></xs:enumeration>
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
  </xs:attribute>
</xs:element>
```



```
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

SetContainer 元素

表 A-129
属性 SetContainer

属性	使用	说明	有效值
source	必须的		string
target	必须的		string

XML 表示

```
<xs:element name="SetContainer">
  <xs:attribute name="source" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[CreateDocumentOutput 元素](#), [CreateInteractiveDocumentBuilder 元素](#), [CreateInteractiveModelBuilder 元素](#), [CreateModelApplier 元素](#), [CreateModelOutput 元素](#)

相关 元素

[SetProperty 元素](#)

SetProperty 元素

表 A-130
属性 SetProperty

属性	使用	说明	有效值
source	必须的		string
target	必须的		string

XML 表示

```
<xs:element name="SetProperty">
  <xs:attribute name="source" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[CreateDocumentOutput](#) 元素, [CreateInteractiveDocumentBuilder](#) 元素,
[CreateInteractiveModelBuilder](#) 元素, [CreateModelApplier](#) 元素, [CreateModelOutput](#)
元素

相关 元素

[SetContainer](#) 元素

SingleFieldChooserControl 元素

表 A-131
属性 SingleFieldChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	必须的		string
showLabel	optional		boolean
storage	optional		string
types	optional		string

XML 表示

```

<xs:element name="SingleFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="storage" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

SingleFieldValueChooserControl 元素

表 A-132
属性 SingleFieldValueChooserControl

属性	使用	说明	有效值
description	optional		string
descriptionKey	optional		string
fieldControl	optional		string
fieldDirection	optional		in out both none partition
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="SingleFieldValueChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
    default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="fieldControl" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="fieldDirection" type="FIELD-DIRECTION" use="optional">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

```
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SpinnerControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

SingleItemChooserControl 元素

表 A-133

属性 SingleItemChooserControl

属性	使用	说明	有效值
catalog	必须的		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="SingleItemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
```

```

</xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
</xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
</xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
</xs:attribute name="catalog" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[MultiItemChooserControl 元素](#)

SpinnerControl 元素

表 A-134

属性 SpinnerControl

属性	使用	说明	有效值
columns	optional		positiveInteger
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
maxDecimalDigits	optional		positiveInteger
minDecimalDigits	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean
stepSize	optional		decimal

XML 表示

```

<xs:element name="SpinnerControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

    <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
    <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:choice>
</xs:sequence>
<xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="showLabel" type="xs:boolean" use="optional"
default="true"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional"
default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="columns" type="xs:positiveInteger" use="optional"
default="5"></xs:attribute>
<xs:attribute name="stepSize" type="xs:decimal" use="optional"
default="1.0"></xs:attribute>
<xs:attribute name="minDecimalDigits" type="xs:positiveInteger" use="optional"
default="1"></xs:attribute>
<xs:attribute name="maxDecimalDigits" type="xs:positiveInteger"
use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#)

SPSSDataFormat 元素

XML 表示

```
<xs:element name="SPSSDataFormat"></xs:element>
```

父 元素[DataFormat 元素](#)**StaticText 元素**

表 A-135

属性 StaticText

属性	使用	说明	有效值
text	optional		string
textKey	optional		string

XML 表示

```

<xs:element name="StaticText">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="text" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="textKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)**子 元素**[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)**相关 元素**

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[ModelViewerPanel 元素](#), [SelectorPanel 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#),
[TextBrowserPanel 元素](#)

StatusCode 元素

表 A-136

属性 StatusCode

属性	使用	说明	有效值
code	必须的		integer
message	optional		string
messageKey	optional		string
status	optional		success warning error

XML 表示

```

<xs:element name="StatusCode">
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="status" use="optional" default="success">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="success"></xs:enumeration>
        <xs:enumeration value="warning"></xs:enumeration>
        <xs:enumeration value="error"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="message" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="messageKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[StatusCodes 元素](#)

StatusCodes 元素

表 A-137
属性 StatusCodes

属性	使用	说明	有效值
defaultMessage	optional		string
defaultMessageKey	optional		string

XML 表示

```

<xs:element name="StatusCodes">
  <xs:sequence>
    <xs:element ref="StatusCode" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="defaultMessage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="defaultMessageKey" type="xs:string"
    use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Module 元素](#), [Run 元素](#)

子 元素

[StatusCode 元素](#)

StatusDetail 元素

Supplementary information about a progress or other conditions.

表 A-138
属性 StatusDetail

属性	使用	说明	有效值
destination	可选的		client tracefile console

XML 表示

```
<xs:element name="StatusDetail" type="STATUS-DETAIL">
  <xs:sequence>
    <xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0"
      maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="Message" type="DIAGNOSTIC-MESSAGE"
          minOccurs="0"></xs:element>
        <xs:element name="Parameter" type="xs:string" minOccurs="0"
          maxOccurs="unbounded"></xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="destination" type="STATUS-DESTINATION" default="client">
    <xs:enumeration value="client"></xs:enumeration>
    <xs:enumeration value="tracefile"></xs:enumeration>
    <xs:enumeration value="console"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

子 元素

Diagnostic 元素

Diagnostic 元素

表 A-139
属性 Diagnostic

属性	使用	说明	有效值
code	必须的		integer
severity	可选的		unknown information warning error fatal
source	可选的		string
subCode	可选的		integer

XML 表示

```
<xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0"></xs:element>
    <xs:element name="Parameter" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
```

```

</xs:sequence>
<xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
<xs:attribute name="subCode" type="xs:integer" default="0"></xs:attribute>
<xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
  <xs:enumeration value="unknown"></xs:enumeration>
  <xs:enumeration value="information"></xs:enumeration>
  <xs:enumeration value="warning"></xs:enumeration>
  <xs:enumeration value="error"></xs:enumeration>
  <xs:enumeration value="fatal"></xs:enumeration>
</xs:attribute>
<xs:attribute name="source" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素

[StatusDetail 元素](#)

子 元素

[Message 元素](#), [Parameter 元素](#)

Message 元素

表 A-140
属性 Message

属性	使用	说明	有效值
lang	可选的		NMTOKEN

XML 表示

```

<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>

```

父 元素

[Diagnostic 元素](#)

Parameter 元素

XML 表示

```

<xs:element name="Parameter" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>

```

父 元素

[Diagnostic 元素](#)

Structure 元素

XML 表示

```
<xs:element name="Structure">
  <xs:sequence>
    <xs:element ref="Attribute" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[PropertyType 元素](#)

子 元素

[Attribute 元素](#)

StructuredValue 元素

A sequence of named values (“attributes”).

XML 表示

```
<xs:element name="StructuredValue" type="STRUCTURED-VALUE">
  <xs:sequence>
    <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"></xs:element>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"></xs:element>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

[ListValue 元素](#), [Attribute 元素](#), [ListValue 元素](#), [Parameter 元素](#), [Attribute 元素](#), [ListValue 元素](#)

子 元素

[Attribute 元素](#)

Attribute 元素

表 A-141
属性 Attribute

属性	使用	说明	有效值
name	必须的		string
value	可选的		string

XML 表示

```
<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"></xs:element>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>
```

父 元素

[StructuredValue 元素](#)

子 元素

[DatabaseConnectionValue 元素](#), [ListValue 元素](#), [ListValue 元素](#), [MapValue 元素](#), [StructuredValue 元素](#), [Value 元素](#)

ListValue 元素

A sequence of values. All values must have the same content type but this is not checked.

XML 表示

```
<xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

父 元素

[Attribute 元素](#)

子 元素

[DatabaseConnectionValue 元素](#), [ListValue 元素](#), [MapValue 元素](#), [StructuredValue 元素](#), [Value 元素](#)

SystemControls 元素

表 A-142
属性 SystemControls

属性	使用	说明	有效值
controlsId	必须的		string

XML 表示

```
<xs:element name="SystemControls">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[ModelViewerPanel 元素](#), [SelectorPanel 元素](#), [StaticText 元素](#), [TabbedPanel 元素](#),
[TextBrowserPanel 元素](#)

Tab 元素

表 A-143
属性 Tab

属性	使用	说明	有效值
helpLink	optional		string
id	optional		string
label	必须的		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string

XML 表示

```

<xs:element name="Tab">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="PropertiesPanel"></xs:element>
      <xs:element ref="ExtensionObjectPanel"></xs:element>
      <xs:element ref="TextBrowserPanel"></xs:element>
      <xs:element ref="ModelViewerPanel"></xs:element>
      <xs:element ref="TabbedPanel"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[Tabs 元素](#)

子 元素

[ExtensionObjectPanel 元素](#), [ModelViewerPanel 元素](#), [PropertiesPanel 元素](#),
[TabbedPanel 元素](#), [TextBrowserPanel 元素](#)

TabbedPanel 元素

表 A-144

属性 TabbedPanel

属性	使用	说明	有效值
style	optional		standard sidebar

XML 表示

```
<xs:element name="TabbedPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Tabs"/></xs:element>
  </xs:sequence>
  <xs:attribute name="style" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="standard"/></xs:enumeration>
        <xs:enumeration value="sidebar"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```

父 元素

[Tab 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Tabs 元素](#), [Visible 元素](#)

相关 元素

[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[ModelViewerPanel 元素](#), [SelectorPanel 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TextBrowserPanel 元素](#)

TableControl 元素

表 A-145

属性 TableControl

属性	使用	说明	有效值
columns	optional		positiveInteger
columnWidths	optional		string
description	optional		string

属性	使用	说明	有效值
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
rows	optional		positiveInteger
showLabel	optional		boolean

XML 表示

```

<xs:element name="TableControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional"
  default="8"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional"
  default="20"></xs:attribute>
  <xs:attribute name="columnWidths" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

CheckBoxControl 元素, CheckBoxGroupControl 元素, ClientDirectoryChooserControl 元素, ClientFileChooserControl 元素, DBConnectionChooserControl 元素, DBTableChooserControl 元素, MultiFieldChooserControl 元素, PasswordBoxControl 元素, PropertyControl 元素, RadioButtonGroupControl 元素, ServerDirectoryChooserControl 元素, ServerFileChooserControl 元素, SingleFieldChooserControl 元素, SingleFieldValueChooserControl 元素, SpinnerControl 元素, TextAreaControl 元素, TextBoxControl 元素

Tabs 元素

表 A-146
属性 Tabs

属性	使用	说明	有效值
defaultTab	optional		nonNegativeInteger

XML 表示

```
<xs:element name="Tabs">
  <xs:sequence>
    <xs:element ref="Tab" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
  <xs:attribute name="defaultTab" type="xs:nonNegativeInteger" use="optional"
    default="0"/></xs:attribute>
</xs:element>
```

父 元素

TabbedPanel 元素, UserInterface 元素

子 元素

Tab 元素

TextAreaControl 元素

表 A-147
属性 TextAreaControl

属性	使用	说明	有效值
columns	optional		positiveInteger
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string

属性	使用	说明	有效值
property	必须的		string
rows	optional		positiveInteger
showLabel	optional		boolean
wrapLines	optional		boolean

XML 表示

```

<xs:element name="TextAreaControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional"
  default="8"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional"
  default="20"></xs:attribute>
  <xs:attribute name="wrapLines" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
</xs:element>

```

父 元素

[PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#)

子 元素

[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

相关 元素

[CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [MultiFieldChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元](#)

素, SingleFieldChooserControl 元素, SingleFieldValueChooserControl 元素, SpinnerControl 元素, TableControl 元素, TextBoxControl 元素

TextBoxControl 元素

表 A-148
属性 TextBoxControl

属性	使用	说明	有效值
columns	optional		positiveInteger
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="TextBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional"
  default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional"
  default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional"
  default="20"></xs:attribute>
</xs:element>
```

父 元素

PropertiesPanel 元素, PropertiesSubPanel 元素

子 元素

[Enabled](#) 元素, [Layout](#) 元素, [Visible](#) 元素

相关 元素

[CheckBoxControl](#) 元素, [CheckBoxGroupControl](#) 元素, [ClientDirectoryChooserControl](#) 元素, [ClientFileChooserControl](#) 元素, [DBConnectionChooserControl](#) 元素, [DBTableChooserControl](#) 元素, [MultiFieldChooserControl](#) 元素, [PasswordBoxControl](#) 元素, [PropertyControl](#) 元素, [RadioButtonGroupControl](#) 元素, [ServerDirectoryChooserControl](#) 元素, [ServerFileChooserControl](#) 元素, [SingleFieldChooserControl](#) 元素, [SingleFieldValueChooserControl](#) 元素, [SpinnerControl](#) 元素, [TableControl](#) 元素, [TextAreaControl](#) 元素

TextBrowserPanel 元素

表 A-149

属性 TextBrowserPanel

属性	使用	说明	有效值
columns	optional		string
container	必须的		string
rows	optional		string
textFormat	必须的		plainText html rtf
wrapLines	optional		boolean

XML 表示

```

<xs:element name="TextBrowserPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="textFormat" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="plainText"/></xs:enumeration>
        <xs:enumeration value="html"/></xs:enumeration>
        <xs:enumeration value="rtf"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="rows" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="columns" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="wrapLines" type="xs:boolean" use="optional"
    default="false"/></xs:attribute>
</xs:element>

```

父 元素[Tab 元素](#)**子 元素**[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)**相关 元素**[ActionButton 元素](#), [ComboBoxControl 元素](#), [ExtensionObjectPanel 元素](#),
[ModelViewerPanel 元素](#), [SelectorPanel 元素](#), [StaticText 元素](#), [SystemControls 元
素](#), [TabbedPanel 元素](#)**ToolBarItem 元素**表 A-150
属性 ToolBarItem

属性	使用	说明	有效值
action	必须的		string
offset	optional		nonNegativeInteger
separatorAfter	optional		boolean
separatorBefore	optional		boolean
showIcon	optional		boolean
showLabel	optional		boolean

XML 表示

```

<xs:element name="ToolBarItem">
  <xs:attribute name="action" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional"
    default="false"></xs:attribute>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional"
    default="0"></xs:attribute>
</xs:element>

```

父 元素[Controls 元素](#)

UserInterface 元素

表 A-151
属性 UserInterface

属性	使用	说明	有效值
actionHandler	optional		任意
frameClass	optional		任意

XML 表示

```
<xs:element name="UserInterface">
  <xs:sequence>
    <xs:element ref="Icons" minOccurs="0"></xs:element>
    <xs:element ref="Controls" minOccurs="0"></xs:element>
    <xs:element ref="Tabs" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="frameClass" use="optional"></xs:attribute>
  <xs:attribute name="actionHandler" use="optional"></xs:attribute>
</xs:element>
```

父 元素

[DocumentOutput](#) 元素, [Extension](#) 元素, [InteractiveDocumentBuilder](#) 元素, [InteractiveModelBuilder](#) 元素, [ModelOutput](#) 元素, [Node](#) 元素

子 元素

[Controls](#) 元素, [Icons](#) 元素, [Tabs](#) 元素

UTF8Format 元素

XML 表示

```
<xs:element name="UTF8Format"></xs:element>
```

父 元素

[FileFormatType](#) 元素

Value 元素

A simple value.

表 A-152
属性 Value

属性	使用	说明	有效值
value	必须的		string

XML 表示

```
<xs:element name="Value" type="SIMPLE-VALUE">
```

```
<xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

父 元素

ListValue 元素, Attribute 元素, ListValue 元素, Parameter 元素, Attribute 元素, ListValue 元素

Values 元素

XML 表示

```
<xs:element name="Values">
  <xs:sequence>
    <xs:element name="Value" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

父 元素

AddField 元素, MissingValues 元素, ChangeField 元素, MissingValues 元素, Field 元素, MissingValues 元素

子 元素

Value 元素

Value 元素

表 A-153
属性 Value

属性	使用	说明	有效值
flagProperty	可选的		trueValue falseValue
value	必须的		string
valueLabel	optional		string

XML 表示

```
<xs:element name="Value" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="valueLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="flagProperty">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="trueValue"></xs:enumeration>
        <xs:enumeration value="falseValue"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```


父 元素[Values 元素](#)**Values 元素**表 A-154
属性 Values

属性	使用	说明	有效值
code	必须的		integer
displayLabel	可选的		string
flagValue	可选的		boolean
value	必须的		string

XML 表示

```

<xs:element name="Values" type="FIELD-VALUE">
  <xs:sequence>
    <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="flagValue" type="xs:boolean"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
</xs:element>

```

父 元素[Field 元素](#)**子 元素**[DisplayLabel 元素](#)**DisplayLabel 元素**

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

表 A-155
属性 DisplayLabel

属性	使用	说明	有效值
lang	可选的		NMTOKEN
value	必须的		string

XML 表示

```

<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0"
  maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>

```

```
<xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>
```

父 元素

[Values 元素](#)

Visible 元素

XML 表示

```
<xs:element name="Visible">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

父 元素

[ActionButton 元素](#), [CheckBoxControl 元素](#), [CheckBoxGroupControl 元素](#), [ClientDirectoryChooserControl 元素](#), [ClientFileChooserControl 元素](#), [ComboBoxControl 元素](#), [DBConnectionChooserControl 元素](#), [DBTableChooserControl 元素](#), [ExtensionObjectPanel 元素](#), [ModelViewerPanel 元素](#), [MultiFieldChooserControl 元素](#), [MultiItemChooserControl 元素](#), [PasswordBoxControl 元素](#), [PropertiesPanel 元素](#), [PropertiesSubPanel 元素](#), [PropertyControl 元素](#), [RadioButtonGroupControl 元素](#), [SelectorPanel 元素](#), [ServerDirectoryChooserControl 元素](#), [ServerFileChooserControl 元素](#), [SingleFieldChooserControl 元素](#), [SingleFieldValueChooserControl 元素](#), [SingleItemChooserControl 元素](#), [SpinnerControl 元素](#), [StaticText 元素](#), [SystemControls 元素](#), [TabbedPanel 元素](#), [TableControl 元素](#), [TextAreaControl 元素](#), [TextBoxControl 元素](#), [TextBrowserPanel 元素](#), [ItemChooserControl 类型](#)

子 元素

[And 元素](#), [Condition 元素](#), [Not 元素](#), [Or 元素](#)

扩展 类型

扩展类型通过添加属性和子元素扩展了 XML 文档中的元素。要使用 XML 文档中的扩展类型，必须为元素指定带有 `xsi:type` 属性的扩展类型。然后您才可以使用扩展类型定义的属性和元素。

ItemChooserControl 类型

表 A-156

属性 ItemChooserControl

属性	使用	说明	有效值
catalog	必须的		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	必须的		string
showLabel	optional		boolean

XML 表示

```

<xs:complexType name="ItemChooserControl" mixed="false">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

扩展[ComboBoxControl 元素](#)**子 元素**[Enabled 元素](#), [Layout 元素](#), [Visible 元素](#)

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。



- action
 - 处理程序, 117
 - 按钮, 135
- Action 元素, 224
- Action 元素, 规范文件, 44
- ActionButton 元素, 225
- ActionButton 元素, 规范文件, 135
- Actions 元素, 225
- Actions 元素, 规范文件, 44
- AddField 元素, 226
- AddField 元素, 规范文件, 72, 77
- AdjustedPropensity 元素, 310
- Algorithm 元素, 306
- Algorithm 元素, 规范文件, 89
- And 元素, 230
- And 元素, 规范文件, 79
- Attribute 元素, 230, 297, 361
- Attribute 元素 (Catalogs), 规范文件, 46
- Attribute 元素, 规范文件, 69
- AutoModeling 元素, 311
- Automodeling 元素, 规范文件, 101
- BinaryFormat 元素, 231
- Bundle 元素, 343
- Bundle 元素, 规范文件, 38
- C++
 - language, 195
 - 帮助程序, 213
- Catalog 元素, 232
- Catalog 元素, 规范文件, 46
- Catalogs 元素, 232
- Catalogs 元素, 规范文件, 46
- Cell 元素, 292
- Cell 元素, 规范文件, 167
- ChangeField 元素, 233
- ChangeField 元素, 规范文件, 74
- CheckBoxControl 元素, 236
- CheckBoxControl 元素, 规范文件, 142
- CheckBoxGroupControl 元素, 237
- CheckBoxGroupControl 元素, 规范文件, 143
- CLEF 中的快捷键, 45
- ClientDirectoryChooserControl 元素, 239
- ClientDirectoryChooserControl 元素, 规范文件, 144
- ClientFileChooserControl 元素, 240
- ClientFileChooserControl 元素, 规范文件, 145
- ColumnControl 元素, 规范文件, 161
- ComboBoxControl 元素, 241
- ComboBoxControl 元素, 规范文件, 145
- Command 元素, 242
- CommonObjects 元素, 243
- CommonObjects 元素, 规范文件, 40
- Condition 元素, 244
- Condition 元素, 规范文件, 80
- Constraint 元素, 246
- Constraint 元素, 规范文件, 105
- Constructors 元素, 247
- Constructors 元素, 规范文件, 109
- Container 元素, 247
- Container 元素, 规范文件, 59
- ContainerFile 元素, 248
- Containers 元素, 270, 289 - 290, 314, 322
- Containers 元素, 规范文件, 59
- ContainerTypes 元素, 248
- ContainerTypes 元素, 规范文件, 43
- Controls 元素, 248
- Controls 元素, 规范文件, 119
- CreateDocument 元素, 249
- CreateDocument 元素, 规范文件, 110
- CreateDocumentOutput 元素, 250
- CreateDocumentOutput 元素, 规范文件, 111
- CreateInteractiveDocumentBuilder 元素, 250
- CreateInteractiveModelBuilder 元素, 251
- CreateInteractiveModelBuilder 元素, 规范文件, 97
- CreateModel 元素, 252
- CreateModel 元素, 规范文件, 110
- CreateModelApplier 元素, 253
- CreateModelApplier 元素, 规范文件, 112
- CreateModelOutput 元素, 254
- CreateModelOutput 元素, 规范文件, 110
- data
 - 类型, 202
 - 记录器节点, 55
 - 读取器节点, 30, 55
 - 转换器节点, 31, 55
- DatabaseConnectionValue 元素, 255
- DataFile 元素, 256
- DataFormat 元素, 256
- DataModel 元素, 256
- DBConnectionChooserControl 元素, 263
- DBConnectionChooserControl 元素, 规范文件, 146
- DBTableChooserControl 元素, 264
- DBTableChooserControl 元素, 规范文件, 148
- DefaultValue 元素, 265
- DefaultValue 元素, 规范文件, 62
- DelimitedDataFormat 元素, 267
- Diagnostic 元素, 273, 358
- DisplayLabel 元素, 268, 303, 373
- DocumentBuilder 元素, 268
- DocumentBuilder 元素, 规范文件, 108
- DocumentGeneration 元素, 269
- DocumentGeneration 元素, 规范文件, 108
- DocumentOutput 元素, 269
- DocumentOutput 元素, 规范文件, 108
- DocumentType 元素, 270
- DocumentType 元素, 规范文件, 44
- Enabled 元素, 271
- Enabled 元素, 规范文件, 173
- Enum 元素, 272
- Enum 元素, 规范文件, 68
- Enumeration 元素, 272
- Enumeration 元素, 规范文件, 68

- ErrorDetail 元素, 273
- Exclude 元素, 规范文件, 78
- Executable 元素, 275
- Execution 元素, 275
- Execution 元素, 规范文件, 61
- ExpertSettings 元素, 312
- ExpertSettings 元素, 规范文件, 103
- extension
 - 对象面板, 129
 - 文件夹, 5
- Extension 元素, 275
- Extension 元素, 规范文件, 36
- extension.xml 文件, 6, 34
- ExtensionDetails 元素, 276
- ExtensionDetails 元素, 规范文件, 37
- ExtensionObjectPanel 元素, 277
- ExtensionObjectPanel 元素, 规范文件, 130
- field
 - 元数据, 77
 - 集合, 77 - 78
- Field 元素, 262, 278
- FieldFormats 元素, 257, 281
- FieldGroup 元素, 260, 282, 284
- FieldGroups 元素, 259, 283
- FieldName 元素, 261, 283, 285
- Fields 元素, 261
- FieldSet 元素, 规范文件, 78
- FileFormatType 元素, 285
- FileFormatTypes 元素, 286
- filesystem, 203
- ForEach 元素, 286
- ForEach 元素, 规范文件, 75, 77
- HelpInfo 元素, 344
- HelpInfo 元素, 规范文件, 182
- HTML 帮助
 - 本地化, 190
 - 链接到, 180
- Icon 元素, 287
- Icon 元素, 规范文件, 118
- Icons 元素, 287
- Identifier 元素, 266
- Include 元素, 规范文件, 78
- InputFields 元素, 307
- InputFields 元素, 规范文件, 90
- InputFiles 元素, 288
- InputFiles 元素, 规范文件, 62
- InteractiveDocumentBuilder 元素, 288
- InteractiveModelBuilder 元素, 289
- InteractiveModelBuilder 元素, 规范文件, 98
- ISO 标准, 语言代码, 186
- ItemChooserControl 类型, 375
- JarFile 元素, 343
- JarFile 元素, 规范文件, 39
- Java, 6
 - API, 5
 - 类, 38 - 39, 44, 66, 117, 129, 153, 178
- JavaHelp
 - 本地化, 190
 - 链接到, 181
- KeyValue 元素, 295
- Layout 元素, 291
- Layout 元素, 规范文件, 166
- License 元素, 293
- ListValue 元素, 293, 298, 362
- MapEntry 元素, 295
- MapValue 元素, 294
- Menu 元素, 298
- Menu 元素, 规范文件, 120
- MenuItem 元素, 300
- MenuItem 元素, 规范文件, 121
- Message 元素, 274, 359
- MissingValues 元素, 228, 234, 279, 301
- ModelBuilder 元素, 304
- ModelBuilder 元素, 规范文件, 87
- ModelDetail 元素, 253
- ModelEvaluation 元素, 309
- ModelField 元素, 229, 235, 280
- ModelFields 元素, 309
- ModelFields 元素, 规范文件, 93
- ModelGeneration 元素, 308
- ModelGeneration 元素, 规范文件, 93
- ModelingFields 元素, 306
- ModelingFields 元素, 规范文件, 89
- ModelOutput 元素, 313
- ModelOutput 元素, 规范文件, 95
- ModelProvider 元素, 315
- ModelProvider 元素, 规范文件, 57
- ModelType 元素, 315
- ModelType 元素, 规范文件, 44
- ModelViewerPanel 元素, 316
- ModelViewerPanel 元素, 规范文件, 133
- Module 元素, 317
- Module 元素, 规范文件, 64
- MultiFieldChooserControl 元素, 317
- MultiFieldChooserControl 元素, 规范文件, 149
- MultiItemChooserControl 元素, 319
- MultiItemChooserControl 元素, 规范文件, 152
- Node 元素, 320
- Node 元素, 规范文件, 54
- nodes, 4
 - 测试 CLEF 扩展, 219
- Not 元素, 322
- Not 元素, 规范文件, 79
- NumberFormat 元素, 258, 281, 323
- NumericInfo 元素, 324
- Option 元素, 324
- OptionCode 元素, 325
- Or 元素, 325
- Or 元素, 规范文件, 79
- output
 - 文件, 4
- OutputDataModel 元素, 326
- OutputDataModel 元素, 规范文件, 66

索引

- OutputFields 元素, 307
 OutputFields 元素, 规范文件, 92
 OutputFiles 元素, 327
 OutputFiles 元素, 规范文件, 63
 Palette 元素, 327
 Palette 元素, 规范文件, 48
 Palettes 元素, 规范文件, 48
 Parameter 元素, 274, 329, 359
 Parameters 元素, 328
 PasswordBoxControl 元素, 330
 PasswordBoxControl 元素, 规范文件, 152
 PMML 格式, 模型输出, 57, 133
 Predictive Server API (PSAPI), 195
 Properties 元素, 331
 Properties 元素, 规范文件, 58
 运行时, 62
 PropertiesPanel 元素, 331
 PropertiesPanel 元素, 规范文件
 嵌套, 140
 用于选项卡和属性子面板, 132
 PropertiesSubPanel 元素, 333
 PropertiesSubPanel 元素, 规范文件, 138
 Property 元素, 334
 Property 元素, 规范文件, 58
 运行时, 62
 PropertyControl 元素, 336
 PropertyControl 元素, 规范文件, 153
 PropertyGroup 元素, 337
 PropertyGroup 元素, 规范文件, 102 - 103
 PropertyMap 元素, 313
 PropertyMapping 元素, 313
 PropertySet 元素, 规范文件, 42
 PropertySets 元素, 337
 PropertySets 元素, 规范文件, 42
 PropertyType 元素, 338
 PropertyType 元素, 规范文件, 41
 PropertyTypes 元素, 339
 PropertyTypes 元素, 规范文件, 41
 PSAPI, 5
 RadioButtonGroupControl 元素, 339
 RadioButtonGroupControl 元素, 规范文件, 154
 Range 元素, 302, 341
 RawPropensity 元素, 310
 RemoveField 元素, 342
 RemoveField 元素, 规范文件, 75
 Resources 元素, 342
 Resources 元素, 规范文件, 38
 Run 元素, 345
 Selector 元素, 346
 SelectorPanel 元素, 345
 ServerDirectoryChooserControl 元素, 346
 ServerDirectoryChooserControl 元素, 规范文件, 156
 ServerFileChooserControl 元素, 348
 ServerFileChooserControl 元素, 规范文件, 156
 ServerTempDir 元素, 266
 ServerTempFile 元素, 266
 SetContainer 元素, 349
 SetProperty 元素, 350
 SharedLibrary 元素, 344
 SharedLibrary 元素, 规范文件, 39
 SimpleSettings 元素, 312
 SimpleSettings 元素, 规范文件, 102
 SingleFieldChooserControl 元素, 350
 SingleFieldChooserControl 元素, 规范文件, 157
 SingleFieldValueChooserControl 元素, 352
 SingleItemChooserControl 元素, 353
 SingleItemChooserControl 元素, 规范文件, 159
 SpinnerControl 元素, 354
 SpinnerControl 元素, 规范文件, 160
 SPSSDataFormat 元素, 355
 SQL pushback, 203
 SQL 生成文档, XML 输出, 211
 StaticText 元素, 356
 StaticText 元素, 规范文件, 136
 StatusCode 元素, 356
 StatusCode 元素, 规范文件, 64, 209
 StatusCodes 元素, 357
 StatusCodes 元素, 规范文件, 64
 StatusDetail 元素, 357
 Structure 元素, 360
 Structure 元素, 规范文件, 69
 StructuredValue 元素, 296, 360
 SystemControls 元素, 362
 SystemControls 元素, 规范文件, 137
 Tab 元素, 363
 Tab 元素, 规范文件, 125
 TabbedPanel 元素, 364
 TableControl 元素, 364
 TableControl 元素, 规范文件, 160
 Tabs 元素, 366
 Tabs 元素, 规范文件, 124
 TextAreaControl 元素, 366
 TextAreaControl 元素, 规范文件, 162
 TextBoxControl 元素, 368
 TextBoxControl 元素, 规范文件, 163
 TextBrowserPanel 元素, 369
 TextBrowserPanel 元素, 规范文件, 128
 ToolbarItem 元素, 370
 ToolbarItem 元素, 规范文件, 123
 UI 组件
 操作按钮, 135
 系统控件, 136
 静态文本, 136
 UserInterface 元素, 371
 UserInterface 元素, 规范文件, 61
 自定义选项板, 48
 UTF8Format 元素, 371
 Value 元素, 303, 371 - 372
 Values 元素, 302, 372 - 373
 VariableImportance 元素, 311
 Visible 元素, 374
 Visible 元素, 规范文件, 178
 Windows 中设置的区域, 185

- XML
 - 声明, 规范文件, 36
 - 解析 API, 215
 - 输出文档, 206
- 与早期版本兼容, 为扩展保持, 85
- 临时文件, 203
 - 服务器, 62
- 为模型构建器节点指定的算法, 89
- 主机信息文档, XML 输出, 209
- 主机函数, API, 198
- 主窗口, 自定义, 123
- 交互式
 - 模型, 构建, 87, 96
- 交互窗口, 96
- 体系结构
 - 服务器端 API, 195
 - 系统, 1
 - 值列表, 46
 - 值列表, 用于枚举属性, 68
 - 值类型, 属性, 67
 - 倾向, 在数据模型中指定, 73, 79
- 元数据, 字段, 77
- 克隆, 模型, 43
- 共享库, 39, 64, 202
- 分发扩展, 223
- 删除
 - 选项板和子选项板, 52
- 加密字符串, 67
- 单字段选择器控件, 157
- 单选按钮组, 154
 - 更改显示顺序, 166
 - 更改行数, 165
- 单项目选择器控件, 159
- 卸载扩展, 223
- 原始倾向, 73
- 参数元素, 状态详细信息文档, 212
- 参数文档, XML 输出, 211
- 句柄, 在回调函数中, 197
- 向后兼容性, 保持, 85
- 商标, 377
- 回调函数, API, 195, 197
- 图像, 为图标创建, 18
- 图形, 44
- 图形要求, 图标, 18
- 图标
 - 创建图像, 18
 - 区域, 对话框, 23
 - 图形要求, 18
 - 已生成的模型, 17
 - 类型, 119
 - 节点, 17
 - 设计, 16
- 块, 模型, 11
- 基于 C 的 API, 5
- 处理流程, 服务器端 API, 199
- 复合条件, 83
- 复选框, 142
- 复选框组, 143
 - 更改显示顺序, 166
 - 更改行数, 165
- 多字段选择器, 149
- 多项目选择器控件, 152
- 子选项板
 - 为节点指定, 15, 48, 55
 - 删除, 52
 - 隐藏, 52
- 字段
 - 组, 93 - 94
- 字符串
 - 加密的, 67
 - 求值的, 71
- 存储类型, 202
- 安装扩展, 223
- 客户端
 - 文件选择器, 145
 - 目录选择器, 144
- 客户端 API, 5, 193
 - 使用, 194
- 类, 193
- 客户端组件, 1
- 容器, 43, 59
 - 内容, 检查, 220
 - 文件, 63
 - 检查内容, 220
 - 类型, 43

索引

- 密码框, 152
- 对等, 195
 - 函数, API, 196
- 对话框, 设计, 20
- 对象定义部分, 规范文件, 53
- 对象标识符, 54
- 屏幕组件的可见性, 控制, 178
- 属性
 - panel, 131
 - 子面板, 138
 - 定义, 58
 - 枚举, 67
 - 检查设置, 220
 - 运行时, 62
 - 键控, 41, 70
 - 面板(嵌套), 140
 - 属性, 控制器, 141
 - 属性, 类型
 - structured, 69
 - 枚举, 68
 - 属性控件, 134
 - PropertyControl 元素, 153
 - UI 组件, 134
 - 属性面板, 138
 - 控制器, 141
 - 属性控件布局
 - 定制, 165
 - 标准, 163
 - 属性文件 (.properties), 186
 - 属性设置, 检查, 220
 - 属性面板控件
 - 属性子面板, 138
 - 属性面板(嵌套), 140
- 工具提示文本, 指定, 26, 45
- 工具栏
 - 区域, 对话框, 23
 - 项目, 自定义, 14, 123
- 已生成对象
 - 图形或报告, 107
 - 模型, 87
- 布局, 属性控件
 - 定制, 165
 - 标准, 163
- 帧类, 117
- 帮助主题, 指定用于显示, 183
- 帮助系统
 - 本地化, 190
 - 的位置, 182
 - 链接到, 180
- 帮助链接, 为节点指定, 56
- 帮助集文件, JavaHelp, 181
- 库, 共享(服务器端), 39, 64, 202
- 应用模型, 107
- 应用程序编程接口(API)
 - PSAPI, 5, 195
 - 基于 C, 5
 - 基于 Java, 5
 - 客户端, 5, 193
 - 文档, 193
 - 服务器端, 5, 195
- 微调控件, 160
- 快捷键
 - 在 CLEF 中, 126
- 执行, 外部(扩展过程), 220
- 执行要求文档, XML 输出, 209
- 扩展, 1
 - 保持向后兼容性, 85
 - 分发, 223
 - 卸载, 223
 - 安装, 223
 - 本地化, 185
 - 模块, 195
- 扩展过程的外部执行, 220
- 报告, 44
- 按钮区域, 对话框, 26
- 挖掘功能, 模型构建器, 88
- 控件, 屏幕属性, 134
 - UI 组件, 134
 - 属性面板, 138
 - 控制器, 141
- 控件, 节点对话框, 20
- 控件顺序, 更改, 166
- 控制器, 141
 - 单字段选择器, 157
 - 单选按钮组, 154
 - 单项目选择器, 159
 - 复选框, 142
 - 复选框组, 143
 - 多字段选择器, 149
 - 多项目选择器, 152
 - 客户端文件选择器, 145
 - 客户端目录选择器, 144
 - 密码框, 152
 - 属性, 141
 - 属性控件, 153
 - 微调框, 160
 - 数据库表选择器, 148
 - 数据库连接选择器, 146
 - 文本区域, 162
 - 文本框, 163
 - 服务器文件选择器, 156
 - 服务器目录选择器, 156
 - 组合框, 145
 - 表, 160

- 提供程序, 数据模型, 77
- 操作, 在规范文件中, 72
- 数据
 - 挖掘功能, 模型构建器, 88
 - 记录器节点, 13
 - 读取器节点, 10
 - 转换器节点, 11
- 数据模型, 4, 207
 - 处理, 203
 - 提供程序, 77
- 数据模型文档, XML 输出, 207
- 整体建模节点, 101
- 文件夹, 扩展, 5
- 文件结构, 5
- 文本
 - 区域控件, 162
 - 框控件, 163
 - 浏览器面板, 127
- 文档, 44, 86
 - 构建, 107
 - 构建器节点, 12, 31, 55, 86, 107
 - 类型, 44
 - 输出, 为节点定义, 108
 - 输出对象, 12
- 服务函数, API, 195
- 服务器
 - 临时文件, 62
 - 为调试更改配置选项, 222
 - 文件选择器控件, 156
 - 目录选择器控件, 156
- 服务器端
 - 库, 39, 64, 202
 - 组件, 2
- 服务器端 API, 5, 195
 - 体系结构, 195
 - 使用, 215
 - 功能, 201
- 本地化
 - 帮助系统, 190
 - 扩展, 185
 - 错误消息, 213
- 条件, 在规范文件中, 80
 - 复合, 83
 - 用于控制屏幕组件的可见性, 178
 - 用于控制显示特征, 173
 - 简单, 83
- 构建
 - 交互模型, 87, 96
 - 模型, 87
- 构造函数, 86
- 构造函数, 使用, 109
- 枚举属性, 67 - 68
- 标签, 放置到组件的上方, 165
- 标题栏, 对话框, 23
- 模块信息文档, XML 输出, 210
- 模块函数, API, 196
- 模块, 扩展, 195
- 模型, 86
 - data, 4
 - 交互式, 96
 - 块, 11
 - 填充器节点, 13, 55, 86, 112
 - 应用, 107
 - 构建, 87
 - 构建器节点, 11, 32, 55, 86 - 87
 - 查看器面板, 133
 - 签名, 93
 - 类型, 44
 - 自动, 101
 - 输出对象, 86
- 模型输出
 - 为节点定义, 95
 - 对象, 11, 86
- 模型选项卡, 管理器窗格, 95
- 求值的字符串, 71
- 法律注意事项, 376
- 注解选项卡, 节点对话框, 25
- 注释行, 在规范文件中, 34
- 测试
 - CLEF 扩展, 219
 - 本地化后的节点和帮助, 191
- 消息元素, 状态详细信息文档, 212
- 状态区域, 对话框, 24
- 状态详细信息文档, XML 输出, 212
- 用户界面
 - 定义, 113
 - 设计, 20
- 用户界面部分, 规范文件, 116
 - 自定义选项板, 48
- 目录, 46
- 示例节点, CLEF, 29
- 签名, 模型, 93
 - “算法设置”对话框, 102 - 103, 105
- 类, 6
 - 客户端 API, 193
- 精确的控件位置, 指定, 166
- 系统
 - 控件, 136

索引

- 菜单, 120
- 组, 字段, 93 - 94
- 组合框, 145
- 结构化属性, 69
- 结构声明, 67
- 缓冲, 数据, 209
- 缓存状态, 节点, 17
- 背景, 图标, 18
- 脚本名称, 54
 - 为属性指定, 58
 - 为节点指定, 56, 84
- 自动建模, 101
- 自定义属性控件布局, 165
 - 简单, 165
 - 高级, 166
- 自定义输出窗口, 178
- 节点, 9
 - 信息文档 (XML), 203
 - 函数, API, 198
 - 名称, 自定义, 26
 - 图标, 设计, 16
 - 定义, 54
 - 属性, 55
 - 数据变换器, 11
 - 数据记录器, 13
 - 数据读取器, 10
 - 整体, 101
 - 文档构建器, 12
 - 模型填充器, 13
 - 模型构建器, 11
 - 类型, 4, 201
 - 缓存状态, 17
- 节点信息文档, XML 输出, 210
- 菜单
 - 区域, 对话框, 23
 - 项目, 自定义, 13, 121
- 菜单, 标准和自定义, 13, 120
- 行, 为复选框和单选按钮组更改行数, 165
- 表
 - 表选择器, 148
 - 连接选择器, 146
- 表控件, 160
- 规范文件, 1, 3, 34
- 角色, 在模型输出中, 79
- 解析, XML, 215
- 访问键, 126
- 评分数据, 26
- 诊断元素, 状态详细信息文档, 212
- 语言
 - 代码, ISO 标准, 186
 - 设置, 185
- 调整倾向, 73
- 调色板
 - 为节点指定, 15, 48, 55
 - 删除, 52
 - 隐藏, 52
- 调试
 - 扩展, 219
 - 更改服务器配置选项, 222
 - “调试”选项卡, “节点”对话框, 37, 220
- 资源, 扩展, 195
- 资源数据包, 38
- 轮廓, 16
- 辅助功能, 185, 191
- 输入文件, 4, 62
- 输入文件的 ContainerFile 元素, 规范文件, 62
- 输出
 - 文件, 62
 - 文档 (XML), 206
- 输出对象
 - 文档, 12
 - 模型, 11
- 输出文件的 ContainerFile 元素, 规范文件, 63
- 输出窗口, 113
 - 定制, 178
 - 设计, 26
- 输出选项卡, 管理器窗格, 108
- 边框, 图标, 17
- 运行时属性, 62
- 进度函数, API, 198
- 迭代, 在规范文件中, 75
- 迭代器函数, API, 198
- 选项卡, 在对话框或窗口上定义, 124
- 选项卡区域, 对话框, 24
- 通道函数, API, 199
- 错误处理, 214
- 错误消息, 本地化, 213
- 错误详细信息文档, XML 输出, 209
- 键控属性, 41, 70
- 键盘快捷键, 126
- 隐藏选项板和子选项板, 52

静态文本, 136

面板

属性子面板, 138

属性面板, 131, 138

扩展对象, 129

指定, 127

文本浏览器, 127

模型查看器, 133

面板区域, 对话框, 24