IBM SPSS Modeler 15 CLEF 開発者ガイド

注 : この情報とサポートされている製品をご使用になる前に、「注意事項」(p.) の一般情報をお読みください。

本版は IBM SPSS Modeler 15 ,および新版で指示されるまで後続するすべてのリリースおよび変更に対して適用されます。

Adobe 製品のスクリーンショットは Adobe Systems Incorporated の許可を得て転載しています。

Microsoft 製品のスクリーンショットは Microsoft 社の許可を得て転載しています。

Licensed Materials - Property of IBM

© Copyright IBM Corporation 1994, 2012.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

はじめに

IBM® SPSS® Modeler は、IBM Corp. が開発した企業強化用のデータ マイニング ワークベンチです。SPSS Modeler を使用すると、企業はデータを詳しく調べることにより顧客および一般市民とのリレーションシップを強化することができます。企業は、SPSS Modeler を使って得られた情報に基づいて利益をもたらす顧客を獲得し、抱き合わせ販売の機会を見つけ、新規顧客を引き付け、不正を発見し、リスクを減少させ、政府機関へのサービスの提供を改善することができます。

SPSS Modeler の視覚的インターフェイスを使用すると、特定ビジネスの専門知識を適用し、より強力な予測モデルを実現し、解決までの時間を短縮します。SPSS Modeler では、予測、分類、セグメント化、および関連性検出アルゴリズムなど、さまざまなモデル作成手法を提供しています。モデルを作成した後は、IBM® SPSS® Modeler Solution Publisher により、企業全体の意思決定者やデータベースにモデルを配布することが可能になります。

IBM Business Analytics について

IBM Business Analytics ソフトウェアは、意思決定者がビジネス パフォーマンスを向上させるために信頼する完全で、一貫した正確な情報を提供します。ビジネス インテリジェンス、予測分析、財務実績および戦略管理、および分析アプリケーションの包括的なポートフォリオを利用することによって、現在の実績を明確、迅速に理解し、将来の結果を予測することができます。豊富な業界のソリューション、実績ある実例、専門サービスと組み合わせ、さまざまな規模の組織が、高い生産性を実現、意思決定を自信を持って自動化し、より良い決定をもたらします。

このポートフォリオの一部として、IBM SPSS Predictive Analytics ソフトウェアを使用する組織は、将来のイベントを予測し、その洞察に基づいて積極的に行動し、より優れた業績を実現することができます。全世界の企業、政府、学術分野のお客様が IBM SPSS の技術を活用し、不正行為を減少させ、リスクを軽減させながら、顧客の獲得、保持、成長において、競争優位を高めることができます。IBM SPSS ソフトウェアを日々の業務に取り入れることによって、組織は業務目標を達成し、大きな競争的優位を獲得することができるよう、意思決定を方向付け、自動化することができるようになります。お問い合わせは、http://www.ibm.com/spss を参照してください。

テクニカル サポート

お客様はテクニカル サポートをご利用いただけます。IBM Corp. 製品の使用方法、または対応するハードウェア環境へのインストールについてサポートが必要な場合は、テクニカル サポートにご連絡ください。テクニカ

© Copyright IBM Corporation 1994, 2012.

ル サポートの詳細は、IBM Corp. Web ページ http://www.ibm.com/support を参照してください。ご本人、組織、サポートの同意を確認できるものをご用意ください。

内容

1 概要 1

CLEF の概要 1
システム アーキテクチャ 1
クライアント側コンポーネント 1
サーバー側コンポーネント 2
CLEF の機能 3
特性ファイル 4
ノード 4
Data Model 4
入力ファイルと出力ファイル 5
アプリケーション プログラミング インターフェイス (API) 5
ファイル構造 6
クライアント側コンポーネント 6
サーバー側コンポーネント 8

2 ノード 10

ノードの概要 10
データ リーダー ノード 12
データ変換ノード 12
モデル ビルダー ノード 12
ドキュメント ビルダー ノード 14
モデル アプライヤ ノード 15
データ ライター ノード 15
メニュー、ツールバーおよびパレット 15
メニューおよびサブメニュー 16
ツールバー 17
パレットおよびサブパレット 17
ノード アイコンのデザイン 18
枠線(O) 19
背景 20
グラフィックの必要条件 21
カスタム画像の作成 21
画像ファイルをノード設定に追加 22
ダイアログ ボックスのデザイン 23
ノードのダイアログ ボックスについて 23

ダイアログ ボックス デザインのガイドライン 24
ダイアログ ボックス コンポーネント 25
出力ウィンドウのデザイン 32

3 CLEF の例 35

サンプルについて 35
サンプルの有効化 35
データ リーダー ノード (Apache Log Reader) 36
データ変換ノード (URL Parser) 37
ドキュメント ビルダー ノード (Web Status Report) 38
モデル ビルダー ノード (Interaction) 38
設定ファイルの検証 40
ソース コードの検証 40
サンプルの削除 41

4 特性ファイル 42

特性ファイルの概要 42
特性ファイルの例 43
XML 宣言 45
Extension 要素 45
Extension Details セクション 45
Resources セクション 46
バンドル 47
Jar ファイル 48
共有ライブラリ 48
ヘルプ情報 49
Common Objects セクション 49
プロパティ タイプ 50
プロパティ セット 52
コンテナ タイプ 53
Actions 55
カタログ 56
User Interface (Palettes) セクション 59
Object Definition セクション 65
オブジェクト識別子 66

モデル ビルダー . . . 70
ドキュメント ビルダー . . . 70
モデル プロバイダ . . . 71
Properties . . . 71
コンテナ . . . 73
ユーザー インターフェイス . . . 75
実行 . . . 75
出力データ モデル . . . 81
Constructors . . . 82
共通機能 . . . 83
値の種類 . . . 83
評価文字列 . . . 88
操作 . . . 88
フィールドおよびフィールド メタデータ . . . 94
フィールド セット . . . 95
役割 . . . 97
論理演算子 . . . 98
Conditions . . . 98
スクリプトの CLEF ノードの使用 . . . 103
下位互換性の保持 . . . 105

5 モデルおよびドキュメントの構築 107

モデルおよびドキュメント構築の概要 . . . 107
[モデル] . . . 107
文書 . . . 108
Constructors . . . 108
モデルの構築 . . . 109
モデル ビルダー . . . 109
モデル出力 . . . 119
インタラクティブ モデルの構築 . . . 120
自動化されたモデル作成 . . . 125
モデルの適用 . . . 134
ドキュメントの構築 . . . 135
ドキュメント ビルダー . . . 135
ドキュメント出力 . . . 136
コンストラクタの使用 . . . 137
モデル出力の作成 . . . 137
ドキュメント出力の作成 . . . 138

インタラクティブ モデル ビルダーの作成 139
モデル アプライヤの作成 140

6 ユーザー インターフェイスの構築 142

ユーザー インターフェイスについて 142
User Interface セクション. 145
アイコン 148
コントロール. 149
メニュー. 150
メニュー項目 151
ツールバー項目 153
例 :メイン ウィンドウへの追加 154
タブ 155
アクセス キーとキーボード ショートカット 157
パネル設定 159
テキスト ブラウザ パネル 159
拡張オブジェクト パネル 162
プロパティ パネル 164
モデル ビューア パネル 166
プロパティ コントロール設定 167
UI コンポーネント コントロール 168
プロパティ パネル コントロール 172
コントローラ 175
プロパティ コントロールのレイアウト 204
標準的なコントロールのレイアウト 204
コントロールのカスタム レイアウト 206
カスタム出力ウィンドウ 223

7 ヘルプ システムの追加 224

ヘルプ システムの種類 224
HTML Help 224
JavaHelp 225
ヘルプ システムの実装 226
ヘルプ システムの場所および種類の定義 226
表示する特定のヘルプ トピックの指定 227

8 ローカライゼーションとアクセシビリティ 230

はじめに . . . 230
ローカライゼーション . . . 230
プロパティ ファイル . . . 231
ヘルプ ファイル . . . 236
ローカライズされた CLEF ノードの検定 . . . 237
アクセシビリティ . . . 238

9 プログラミング 240

CLEF ノードのプログラミングについて . . . 240
CLEF API ドキュメンテーション . . . 240
クライアント側 API . . . 241
クライアント側 API クラス . . . 241
クライアント側 API の使用方法 . . . 241
予測サーバー API (PSAPI) . . . 242
サーバー側 API . . . 242
アーキテクチャ . . . 242
サービス関数 . . . 243
コールバック関数 . . . 245
プロセス フロー . . . 247
サーバー側 API の機能 . . . 250
エラー処理 . . . 267
XML API の解析 . . . 268
サーバー側 API の使用方法 . . . 268
サーバー側のプログラミング ガイドライン . . . 268

10 テストと配布 274

CLEF 拡張のテスト . . . 274
CLEF 拡張のテスト . . . 274
CLEF 拡張のデバッグ . . . 275
CLEF 拡張の配布 . . . 278
CLEF 拡張のインストール . . . 279
CLEF 拡張のアンインストール . . . 279

付録

A CLEF XML スキーマ 280

CLEF 要素の参照 . . . 280
Action 要素 . . . 280
ActionButton 要素 . . . 281
Actions 要素 . . . 282
AddField 要素 . . . 282
And 要素 . . . 287
Attribute 要素 . . . 287
BinaryFormat 要素 . . . 288
Catalog 要素 . . . 288
Catalogs 要素 . . . 289
ChangeField 要素 . . . 289
CheckBoxControl 要素 . . . 293
CheckBoxGroupControl 要素 . . . 294
ClientDirectoryChooserControl 要素 . . . 296
ClientFileChooserControl 要素 . . . 297
ComboBoxControl 要素 . . . 298
Command 要素 . . . 300
CommonObjects 要素 . . . 300
Condition 要素 . . . 301
Constraint 要素 . . . 303
Constructors 要素 . . . 304
Container 要素 . . . 305
ContainerFile 要素 . . . 305
ContainerTypes 要素 . . . 306
Controls 要素 . . . 306
CreateDocument 要素 . . . 307
CreateDocumentOutput 要素 . . . 307
CreateInteractiveDocumentBuilder 要素 . . . 308
CreateInteractiveModelBuilder 要素 . . . 309
CreateModel 要素 . . . 310
CreateModelApplier 要素 . . . 311
CreateModelOutput 要素 . . . 312
DatabaseConnectionValue 要素 . . . 313
DataFile 要素 . . . 314
DataFormat 要素 . . . 314
DataModel 要素 . . . 315
DBConnectionChooserControl 要素 . . . 322

DBTableChooserControl 要素 323
DefaultValue 要素 324
DelimitedDataFormat 要素 326
DisplayLabel 要素 327
DocumentBuilder 要素 327
DocumentOutput 要素 328
DocumentType 要素 330
Enabled 要素 330
Enumeration 要素 331
ErrorDetail 要素 332
Executable 要素 334
Execution 要素 335
Extension 要素 335
ExtensionDetails 要素 336
ExtensionObjectPanel 要素 337
Field 要素 337
FieldFormats 要素 341
FieldGroup 要素 342
FieldGroups 要素 343
FileFormatType 要素 345
FileFormatTypes 要素 346
ForEach 要素 346
Icon 要素 347
Icons 要素 348
InputFiles 要素 348
InteractiveDocumentBuilder 要素 349
InteractiveModelBuilder 要素 350
Layout 要素 352
License 要素 354
ListValue 要素 354
MapValue 要素 355
Menu 要素 360
MenuItem 要素 361
MissingValues 要素 363
ModelBuilder 要素 366
ModelOutput 要素 375
ModelProvider 要素 377
ModelType 要素 377
ModelViewerPanel 要素 378
Module 要素 379
MultiFieldChooserControl 要素 380
MultiItemChooserControl 要素 381
Node 要素 382

Not 要素 385
NumberFormat 要素 385
NumericInfo 要素 386
Option 要素 387
OptionCode 要素 388
Or 要素 388
OutputDataModel 要素 389
OutputFiles 要素 390
Palette 要素 390
Parameters 要素 391
PasswordBoxControl 要素 393
Properties 要素 394
PropertiesPanel 要素 395
PropertiesSubPanel 要素 396
Property 要素 398
PropertyControl 要素 400
PropertyGroup 要素 401
PropertySets 要素 401
PropertyType 要素 402
PropertyTypes 要素 403
RadioButtonGroupControl 要素 403
Range 要素 405
Range 要素 406
RemoveField 要素 406
Resources 要素 406
Run 要素 409
SelectorPanel 要素 410
ServerDirectoryChooserControl 要素 411
ServerFileChooserControl 要素 412
SetContainer 要素 414
SetProperty 要素 414
SingleFieldChooserControl 要素 415
SingleFieldValueChooserControl 要素 416
SingleItemChooserControl 要素 418
SpinnerControl 要素 419
SPSSDataFormat 要素 420
StaticText 要素 421
StatusCode 要素 421
StatusCodes 要素 422
StatusDetail 要素 423
Structure 要素 425
StructuredValue 要素 425
SystemControls 要素 428

Tab 要素 429
TabbedPanel 要素 429
TableControl 要素 430
Tabs 要素 432
TextAreaControl 要素 432
TextBoxControl 要素 434
TextBrowserPanel 要素 435
ToolbarItem 要素 436
UserInterface 要素 437
UTF8Format 要素 437
Value 要素 437
Values 要素 438
Values 要素 439
Visible 要素 440
拡張 タイプ 441

B 注意事項 443

索引 446

章

1

概要

CLEF の概要

Component-Level Extension Framework（CLEF）は、ユーザーが提供した拡張を IBM® SPSS® Modeler の標準機能に追加できるメカニズムです。拡張には、たとえばデータ処理ルーチンやモデル作成アルゴリズムなどのユーザー独自のプログラムを、メニューの新規エントリまたはノード パレットの新規ノードとして SPSS Modeler に追加し、使用することができる共有ライブラリが含まれます。

これを実行するために、SPSS Modeler では名前、渡す必要のあるコマンド パラメータ、SPSS Modeler のプログラムへのオプションおよびユーザーへの結果の表示方法など、カスタム プログラムについての詳細が必要です。この情報を提供するために、**特性ファイル**と呼ばれる XML 形式のファイルを提供します。SPSS Modeler は、このファイルの情報をメニューの新規エントリまたはノード定義に変換します。

以下に、CLEF 使用の利点を示します。

- エンジニア、コンサルタント、エンド ユーザーが新機能を SPSS Modeler に統合できる、使いやすく、柔軟性の高い強力な環境が提供されます。
- 拡張モジュールがネイティブの SPSS Modeler モジュールと同じ外観で同じ動作ができるようにします。
- 拡張ノードを、ネイティブの SPSS Modeler ノードとできる限り同じスピードおよび効率で実行できるようにします。

システム アーキテクチャ

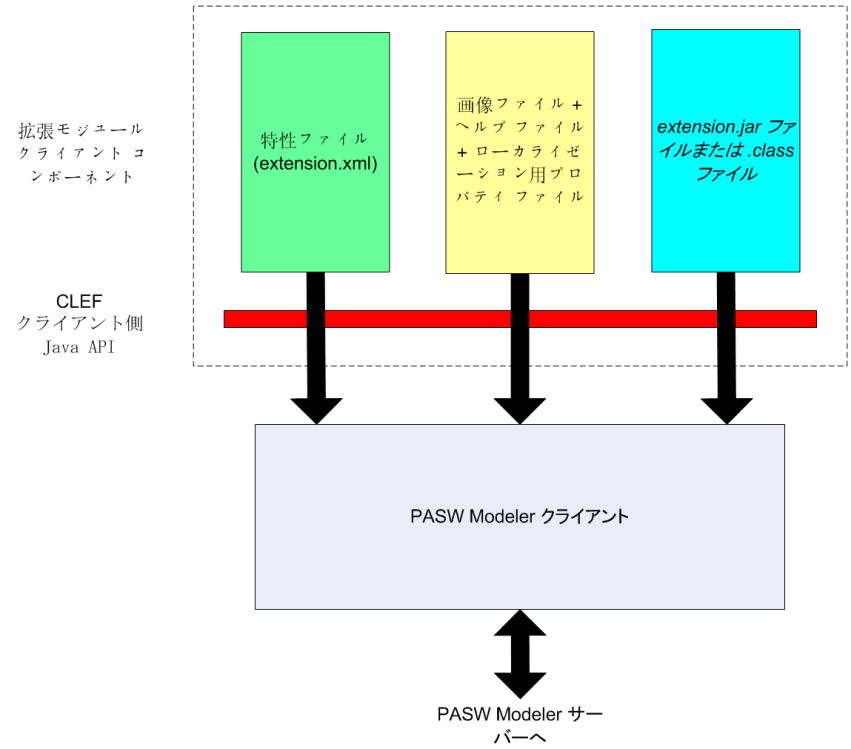
IBM® SPSS® Modeler 自体と同様、 CLEF では 2 層のクライアント/サーバー アーキテクチャを使用し、層は同じマシンまたは 2 つの異なるマシンに存在します。

クライアント側コンポーネント

クライアント層のコンポーネントを次に示します。

© Copyright IBM Corporation 1994, 2012.

図 1-1
クライアント側コンポーネント

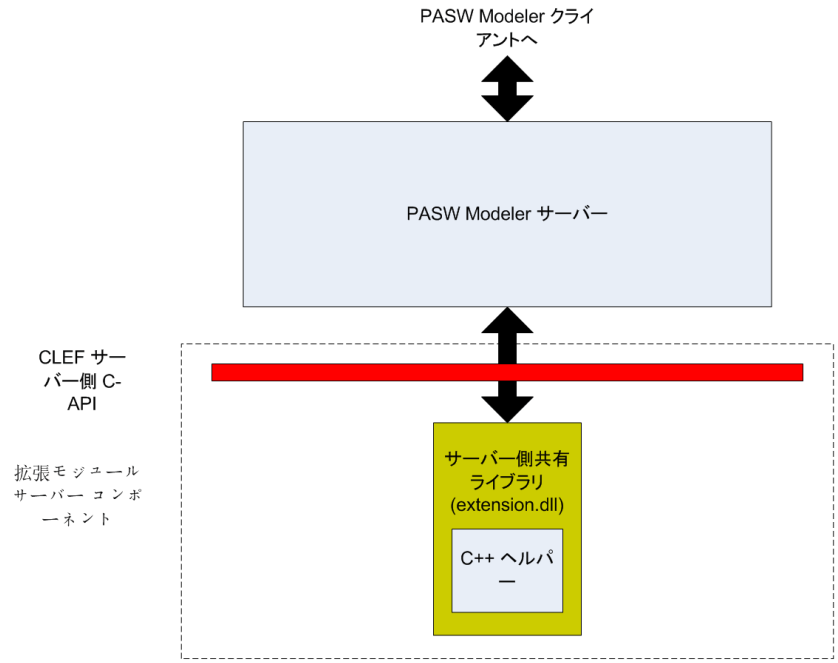


- **特性ファイル：** 拡張で定義されたプロパティ、形式、データ モデルの変更、コントロール、その他の特性を表示します。
- **画像ファイル：** 拡張のノードを識別するために使用された画像が含まれます。
- **ヘルプ ファイル：** 拡張に関するヘルプ情報の表示に使用されます。
- **プロパティ ファイル：** 拡張によって画面に表示される名前、ラベル、メッセージで構成されるテキスト文字列が含まれます。
- **Java の jar または .class ファイル：** 拡張で使用された Java リソースが含まれます。
- **Java アプリケーション プログラミング インターフェイス (API)：** 追加のコントロール、ユーザー インターフェイス コンポーネントを必要とする拡張によって使用できます。または特性ファイルによって直接対話的には提供されません。

サーバー側コンポーネント

サーバー層のコンポーネントを次に示します。

図 1-2
サーバー側コンポーネント



- **共有ライブラリの C ベース API :** 設定および実行設定の取得、これらの設定の持続、実行フィードバック、ジョブ制御（実行の中止など）、SQL 生成、返されたオブジェクトなどの側面をカバーします。
- **サーバー側共有ライブラリ :** ノード拡張をサポートするダイナミック リンク ライブラリ (DLL)。C++ ヘルパーは、ソース コードとして提供され、C++ CLEF モジュールに容易にコンパイルできる C ベースの API のラッパーです。

CLEF の機能

以下の項で、CLEF の多くの主要な機能について説明します。

- 特性ファイル
- ノード
- データ モデル
- 入力ファイルと出力ファイル
- アプリケーション プログラミング インターフェイス (API)

特性ファイル

CLEF 特性ファイルは、新しい拡張の動作を記述した構造仕様を格納する XML ファイルです。特性ファイルは、以下について説明しています。

- 拡張に必要な共有リソース（ローカライズされたテキスト バンドルおよびサーバー側の共有ライブラリなど）。
- ファイル形式またはプロパティの種類など、一般的な定義。
- ノードや出力モデルなど、エンド ユーザーが使用できる新規オブジェクト。

IBM® SPSS® Modeler の起動時、特性ファイルは格納場所からロードされ、ファイルに定義された機能はすぐに使用できます。

詳細は、4 章の「特性ファイル」（ p.42 ）を参照してください。

ノード

新規ノードを実装する IBM® SPSS® Modeler に拡張を追加する場合、まず作成するノードの種類を決定する必要があります（たとえば、ノードがモデルを生成するかデータを転送するのみか）。 詳細は、 2 章 p.10 ノードの概要 を参照してください。

特性ファイルおよび必要な Java クラス、共有ライブラリを作成した後、ファイルを SPSS Modeler が読み込むことができる特定の場所にコピーします。次に SPSS Modeler を起動すると、新規ノードが適切なパレットに追加され、読み込んで使用できます。

Data Model

データ モデルは、IBM® SPSS® Modeler ストリーム内を流れるデータの構造を表しています。ストリーム中の特定の位置のデータを記述するモデルは、データ型ノードに表示される情報に対応しています。ストリーム内にある特定のポイントの既存フィールドの名前を表示し、それらのデータ型を説明します。

SPSS Modeler にノードを追加する場合、ノードに渡されるデータ モデルがそのノードの動作にどのように影響するかを検討してください。たとえば、フィールド作成ノードは、入力データ モデルを取得し、そのデータ モデルに新規フィールドを追加します。次に、SPSS Modeler ストリーム内の次のノードに渡す出力データ モデルを生成します。対照的に、グラフ ノードは、データが他の後続ノードへ渡されないので入力データ モデルを取得します。後続のノードが、どのフィールドが使用可能であるかという正確な情報を示すために、SPSS Modeler はデータ モデルに何が起こるのかを知る必要があります。特性ファイル内にあるデータ モデル情

報は、ストリーム全体にわたってデータ モデルの一貫性を保つために必要な情報を SPSS Modeler に提供しています。

データが入力されるのか、出力されるのか、またはノードを通過するのかに応じて、入力、出力、または両方のデータ モデルを特性ファイルに記述する必要があります。CLEF ノードは、ノードに渡されるフィールドに新規フィールドを追加するか、またはノードに入力されるフィールドを、プログラム自体が生成したノードと置換することによって、データ モデルに影響を与えることができます。特性ファイルの OutputDataModel 要素は、 データモデルの CLEF ノードの影響を説明します。 詳細は、 4 章 p.81 出力データ モデル を参照してください。

入力ファイルと出力ファイル

CLEF ノードが実行される前に生成される一時ファイルを指定できます。サーバーのノード実行への入力であるため、これらのファイルは**入力ファイル**と呼ばれます。たとえば、モデル ビルダー ノードには、内容がノードの実行時に指定の入力ファイルに転送されるモデル コンテナがあります。 詳細は、 4 章 p.77 入力ファイル を参照してください。

モデル ビルダーまたはドキュメント ビルダー ノード実行の結果など、その他の一時ファイルは、サーバーの実行中に生成されます。これらは**出力ファイル**と呼ばれ、ノード実行の後にクライアントに返されます。 詳細は、 4 章 p.78 出力ファイル を参照してください。

アプリケーション プログラミング インターフェイス (API)

実行する拡張によって、アプリケーション プログラミング インターフェイス (API) を利用する必要があります。単純なデータ転送の場合、特性ファイル全体の必要な処理を定義できる場合があります。ただし、詳細な必要条件については、以下の使用可能な API とのインターフェイスが必要です。

- CLEF クライアント側 API
- CLEF サーバー側 API
- 予測サーバー API (PSAPI)

CLEF **クライアント側 API** は、追加のコントロール、ユーザー インターフェイス コンポーネント、または特性ファイルで直接提供されていない双方向性が必要な拡張によって使用できる Java API です。

CLEF **サーバー側 API** は、設定および実行設定の取得、これらの設定の持続、実行フィードバック、ジョブ制御 (実行の中止など)、SQL 生成、返されたオブジェクトなどの側面をカバーする C ベース API です。

予測サーバー API は、データ マイニングおよび予測分析機能を必要とするアプリケーションが使用するための IBM® SPSS® Modeler 機能を公開する Java API です。

詳細は、9 章 「プログラミング」 （ p.240 ） を参照してください。

ファイル構造

CLEF の拡張は、2 つのコンポーネントのセットで構成されています。

- クライアント側コンポーネント
- サーバー側コンポーネント

クライアント側コンポーネントは、拡張設定ファイル、Java クラスおよび .jar ファイル、ローカライズ可能なリソース、画像およびヘルプ ファイルを含むプロパティ バンドルで構成されています。

サーバー側コンポーネントは、拡張ノードが実行される場合に必要な共有ライブラリおよび DLL です。

クライアント側コンポーネント

クライアント側コンポーネントは、IBM® SPSS® Modeler インストール ディレクトリの \ext\lib フォルダにインストールされます。クライアント側コンポーネントの内容は、次のとおりです。

- 特性ファイル
- Java クラスおよび .jar ファイル
- プロパティ ファイル
- 画像ファイル
- ヘルプ ファイル

拡張フォルダ

それぞれの拡張は、\ext\lib 直下の**拡張フォルダ**に配置されます。

拡張フォルダ名の表記方法は、以下をお勧めします。

```
providerTag.id
```

この場合、providerTag は、特性ファイルのExtensionDetails 要素のプロバイダ識別子で、id は同じ要素の拡張子です。

このため、たとえば ExtensionDetails は次のように開始されます。

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

拡張フォルダ名 myco.sorter が使用されます。

特性ファイル

特性ファイル自体の名前は extension.xml となり、拡張サブフォルダの最上部に格納される必要があります。このため前述の例では、特性ファイルへのパスは IBM® SPSS® Modeler インストール ディレクトリの下が次のようになります。

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

Java クラスおよび .jar ファイル

クライアント側 Java API を使用する拡張には、コンパイルされた Java コードが含まれます。このコードは、.class ファイルのセットとして残し、または .jar ファイルとしてパッケージ化できます。

Java .class ファイルは、最上レベルの拡張フォルダに関連して配置されます。たとえば、ActionHandler インターフェイスを実装するクラスのパスは、次のとおりです。

com.my_example.my_extension.MyActionHandler

この場合、.class ファイルは SPSS Modeler インストール ディレクトリの下の次の場所にあります。

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

.jar ファイルは、拡張フォルダの下の任意の場所に置くことができます。特性ファイルの JarFile 要素によって、.jar ファイルの実際の場所を指定します。たとえば、拡張で以下のパスを持つ .jar ファイルを使用する場合

\extension_folder\lib\common-utilities.jar

特性ファイルでは、Resources 要素に次のエントリがあります。

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

詳細は、 4 章 p. 48 Jar ファイル を参照してください。

プロパティ ファイル

ローカライズされたリソース（画面のテキスト、エラー メッセージ、およびそれらの外国語翻訳）は拡張子 .properties を持つファイルに保存され、拡張フォルダの任意の場所に配置できます。 詳細は、 8 章 p. 231 プロパティ ファイル を参照してください。

画像およびヘルプ ファイル

アイコン表示用のグラフィック イメージを含むファイル、ヘルプ システムを含むファイルは、拡張フォルダの下に配置できます。画像ファイルとヘルプ ファイルを、それぞれのサブフォルダに分けておくと便利です。

特性ファイルの Icon 要素の imagePath 属性によって、画像ファイルの場所を宣言します。 詳細は、 6 章 p. 148 アイコン を参照してください。

同様に、特性ファイルの HelpInfo 要素の path 属性を使用して、ヘルプ システムの場所を宣言します。 詳細は、 7 章 p. 226 ヘルプ システムの場所および種類の定義 を参照してください。

例

これらのコンポーネントに基づいたクライアント側のファイル構造は、次のようになります。

```
\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
```

サーバー側コンポーネント

実行に必要な共有ライブラリは、IBM® SPSS® Modeler インストール ディレクトリの \ext\bin フォルダの下にあるフォルダに置く必要があります。次に例を示します。

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

共有ライブラリは、\ext\bin フォルダに直接置くことはできません。

実行時に直接 SPSS Modeler が起動する共有ライブラリの場合、特性ファイルの SharedLibrary 要素で場所を宣言します。 詳細は、 4 章 p. 48 共有ライブラリ を参照してください。

メインの共有ライブラリでは、他のライブラリの使用が必要です。また、メインの共有ライブラリと同じ場所に従属共有ライブラリを配置して、メイン ライブラリが従属ライブラリを検出できるようにする必要があります。

例

サーバー側のファイル構造は次のようになります。

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

章

2

ノード

ノードの概要

新規ノードを実装する拡張子を作成する場合、IBM® SPSS® Modeler ノードの特性を理解する必要があります。ノードの特性を理解することによって、設定ファイルのノードを正しく定義できます。

SPSS Modeler ノードはその機能によって、入力、プロセス、出力、およびモデル作成ノードに分類されます。CLEF において、ノードは少し異なる方法で分類されます。次の表では、2 つのシステム間のマッピングを示しています。

テーブル 2-1
CLEF のノードの種類

SPSS Modeler 分類	パレット	CLEF ノードの種類
入力ノード	入力	データ リーダー
プロセス ノード	レコード設定	データ変換
	フィールド設定	
出力ノード	グラフ作成	ドキュメント ビルダー
	出力（レポート ノード）	
	エクスポート	データ ライター
モデル作成ノード	モデル作成	モデル ビルダー

CLEF ノードを新規作成する場合、CLEF ノードの種類の 1 つとして定義します。選択するノードの種類は、ノードの主要な機能によって異なります。

テーブル 2-2
ノードの種類および機能

CLEF ノードの種類	説明	対応するノード パレット	アイコンの形状
データ リーダー	異なる形式のデータを SPSS Modeler にインポートします。	入力	
データ変換	SPSS Modeler からデータを取得、データを何らかの方法で変更し、変更されたデータを SPSS Modeler ストリームに戻します。	レコード設定、フィールド設定	

© Copyright IBM Corporation 1994, 2012.

CLEF ノードの種類	説明	対応するノードパレット	アイコンの形状
モデル ビルダー	モデルを SPSS Modeler のデータから生成します。	モデル作成	
ドキュメント ビルダー	グラフまたはレポートを SPSS Modeler のデータから生成します。	グラフ作成	
		出力（レポートノード）	
モデル アプライヤ（「モデル ナゲット」）	SPSS Modeler 領域に戻される生成されたモデルのコンテナを定義します。	-	
データ ライター	SPSS Modeler 形式のデータを他のアプリケーションでの使用に適した形式にエクスポートします。	エクスポート	

ノードの種類を、他の属性とともに、設定ファイルの Node 要素に定義します。

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

palette 属性は、ユーザーがノードにアクセスできる SPSS Modeler メインウィンドウにパレットを定義します。 この場合は、[レコード設定] パレットです。この属性を省略すると、ノードは [フィールド設定] パレットに表示されます。

多くのサンプル ノードが IBM® SPSS® Modeler で提供されています。 詳細は、 3 章 p. 35 サンプルについて を参照してください。

データ リーダー ノード

データ リーダー ノードを使用すると、外部ソースのデータを IBM® SPSS® Modeler ストリームに読み込むことができます。SPSS Modeler の［入力］パレットのノードは、データ リーダー ノードと同等のもので、 丸いアイコンの形状によって識別されます。

データ リーダー ノードの設定には、次の詳細が含まれます。

- データ ソース（ファイルまたはデータベースなど）
- レコードの処理（前後のスペースまたはレコードの区切りとして使用する文字の処理など）
- レコード フィールドの除外の有無
- 各フィールドに関連するデータの種類（範囲型、セット型、フラグ型）およびストレージのデータ型（文字列、整数、実数）
- 入力データ モデルの変更の有無

データ リーダー ノードは、ロジックを使用して、ソース データ レコードを読み込むことができます。また、SPSS Modeler のデータ型ノードを使用してより下流で実行することもできます。

サンプルのデータ リーダー ノードは IBM® SPSS® Modeler で提供されています。 詳細は、 3 章 p.35 サンプルについて を参照してください。

データ変換ノード

データ変換ノードは、IBM® SPSS® Modeler ストリームからデータを取得、何らかの方法でデータを変更し、変更したデータをストリームに戻します。SPSS Modeler の［レコード設定］および［フィールド設定］パレットのノードはデータ変換ノードで、六角形のアイコンの形状で識別されます。

データ変換ノードの設定には、次の詳細が含まれます。

- レコードまたフィールドのどちらを変換するか
- データの変更方法

サンプルのデータ変換ノードは IBM® SPSS® Modeler で提供されています。 詳細は、 3 章 p.35 サンプルについて を参照してください。

モデル ビルダー ノード

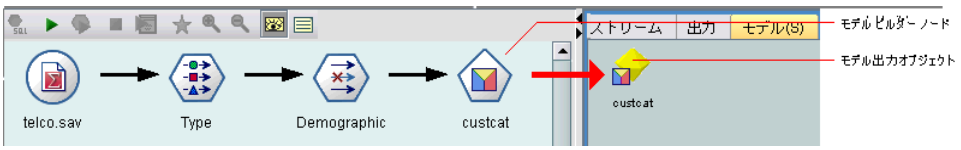
IBM® SPSS® Modeler のモデル構築の概要は、『SPSS Modeler 15 アプリケーション ガイド』の「モデル作成の概要」を参照してください。.

モデル ビルダー ノードは、SPSS Modeler メイン ウィンドウのマネージャ領域の［モデル］または［出力］タブに表示されるオブジェクトを生成します。

SPSS Modeler の［モデル］パレットのノードは、モデル ビルダー ノードの例で、 五角形のアイコンの形状によって識別されます。

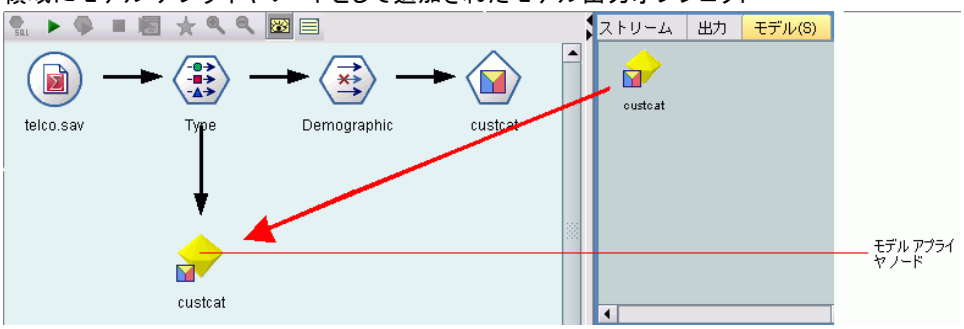
実行すると、モデル ビルダー ノードは［モデル］タブに**モデル出力オブジェクト**（モデル ナゲット）を生成します。

図 2-1
モデル出力オブジェクト（モデル ナゲット）を生成するモデル ビルダー ノード



生成されたモデルが領域に追加されると、モデル アプライヤ ノードの形式となります。

図 2-2
領域にモデル アプライヤ ノードとして追加されたモデル出力オブジェクト



モデル ビルダー ノードの設定には、次が含まれています。

- モデルの生成に使用され、入力および出力フィールドがモデルによるデータのスコアリングに使用されるアルゴリズムなど、モデル構築の詳細
- モデルに使用されるプロパティ
- 出力プロジェクトの保有に使用されるコンテナ
- ノードのダイアログ ボックスのユーザー インターフェイス
- ノードが実行される場合に使用されるプロパティおよびファイル
- 入力データ モデルがノードの実行によって受ける影響
- ノードの実行によって作成されるモデル出力オブジェクトの識別子、その他のオブジェクト
- モデル アプライヤ ノードの識別子（「モデル アプライヤ ノード」（p. 15 ）を参照）

注 ： モデル ビルダー ノードを定義する場合、同じ設定ファイルの他の場所に実際のモデル出力オブジェクトおよびモデル アプライヤ ノードを定義します。

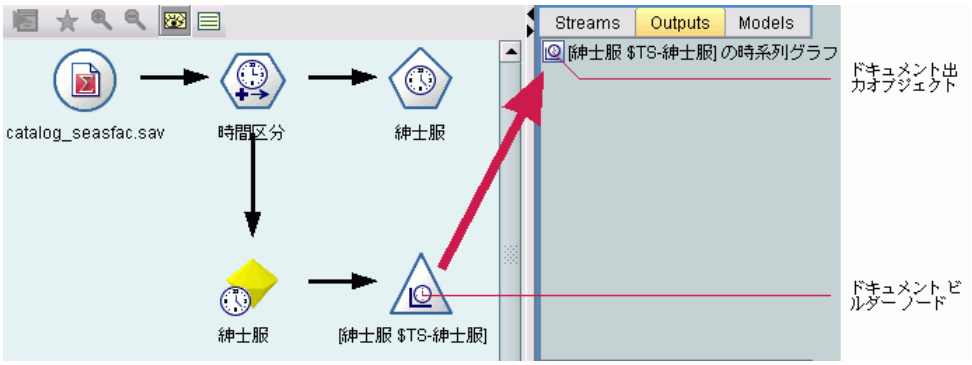
サンプルのモデル ビルダー ノードは IBM® SPSS® Modeler で提供されています。 詳細は、 3 章 p.35 サンプルについて を参照してください。

ドキュメント ビルダー ノード

ドキュメント ビルダー ノードは、IBM® SPSS® Modeler メイン ウィンドウのマネージャ領域の［出力］タブに表示されるオブジェクトを生成します。［グラフ］パレットのノードは、ドキュメント ビルダー ノードの例で、 三角形のアイコンの形状によって識別されます。

実行すると、ドキュメント ビルダー ノードはマネージャ領域の［出力］タブに**ドキュメント出力オブジェクト**を生成します。

図 2-3
ドキュメント出力オブジェクトを生成するドキュメント ビルダー ノード



モデル 出力 オブジェクトとは対照的に、ドキュメント出力オブジェクトを、SPSS Modeler 領域に戻すできません。

ドキュメント ビルダー ノードの設定には、次が含まれています。

- ドキュメント生成コントロールを含むノードのダイアログ ボックスなど、ドキュメント構築の詳細
- ドキュメントに使用されるプロパティ
- 出力プロジェクトの保有に使用されるコンテナ
- ノードのダイアログ ボックスのユーザー インターフェイス
- ノードが実行される場合に使用されるプロパティおよびファイル
- ノードの実行によって作成されるドキュメント出力オブジェクトの識別子、その他のオブジェクト

注 ： ドキュメント ビルダー ノードを定義する場合、同じ設定ファイルの他の場所に実際のドキュメント出力オブジェクトを定義します。

モデル アプライヤ ノード

モデル アプライヤ ノードは、モデルがマネージャ領域の［モデル］タブから IBM® SPSS® Modeler 領域に追加される場合に使用する生成されたモデルのコンテナを定義します。

モデル アプライヤ ノードの設定には、次の詳細が含まれます。

- モデルのコンテナ（テキストおよび HTML など、モデル出力が複数の形式で作成する場合、コンテナは複数）
- ユーザーが［モデル］タブのアプライヤ ノードを参照し、領域上で開く場合に表示されるダイアログ ボックスのユーザー インターフェイスの詳細
- 出力データのモデル
- ノードを含むストリームが実行される場合に行う処理
- ノードを含むストリームが実行される場合に作成されるオブジェクトを処理するコンストラクタ

データ ライター ノード

データ ライター ノードは、IBM® SPSS® Modeler 形式のデータを他のアプリケーションでの使用に適した形式にエクスポートします。SPSS Modeler の［エクスポート］パレットのノードはデータ ライター ノードで、四角形のアイコンの形状で識別されます。

データ ライター ノードの設定には、次が含まれています。

- ストリーム データが書き込まれるファイルまたはデータベースの詳細
- 外部アプリケーションに組み込むことができるよう、ストリーム全体を公開するかどうか（オプション）

メニュー、ツールバーおよびパレット

ユーザーは、IBM® SPSS® Modeler メニュー、ツールバーまたはパレットから拡張子にアクセスできます。拡張子はノードを実装または指定された動作を実行できます。

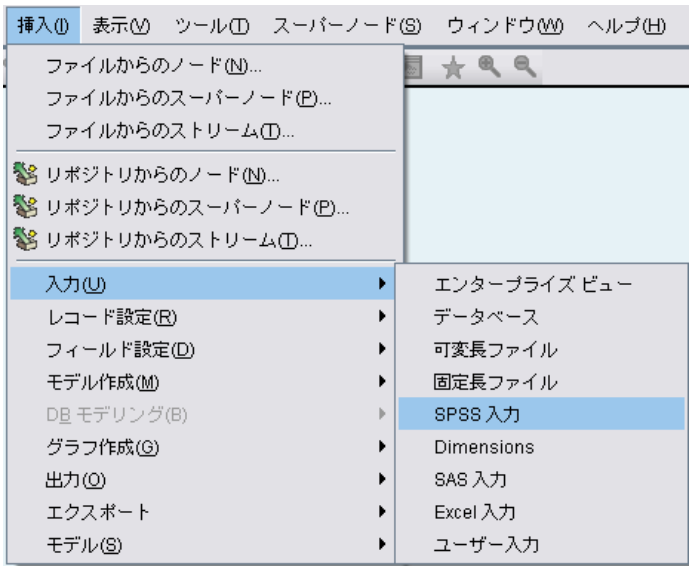
明示的に指定されたメニューからアクセスできる拡張子（ノードまたは動作）は、ツールバーからもアクセス可能にすることができ、またツールバーからアクセス可能な拡張子をメニューからアクセスできるようにすることもできます。

パレットからアクセスできるノードは、自動的に［挿入］メニューの対応する項目からアクセスできます。

メニューおよびサブメニュー

ユーザーは、[挿入] メニューから標準 IBM® SPSS® Modeler ノードにアクセスできます。[挿入] メニューの最後のグループの各項目には（[モデル] とは別に）、一連の関連するノードへのアクセスを提供するサブメニューがあります。

図 2-4
メニューの挿入



これらの項目は、ノード パレットのエントリに直接対応します。パレットにノードを追加すると、ノードを [挿入] メニューの対応するグループに自動的に追加します。

拡張子がノード全体にアクセスできない動作を定義する場合、次のいずれかまたは複数を追加して拡張子を使用できるようにします。

- 新しい項目をシステム メニューまたはサブメニューに追加
- 新しいメニューを SPSS Modeler に追加
- 新しい項目をツールバーに追加（「ツールバー」（ p. 17 ）を参照）

新しいメニューまたはメニュー項目は、オプションで拡張子に関連するアイコンを [挿入] メニュー項目に表示できます。

詳細は、「メニュー」（ p. 150 ）および「メニュー項目」（ p. 151 ）を参照してください。

ツールバー

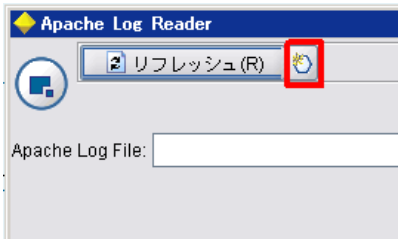
拡張子がノード全体にアクセスできない動作を定義する場合、メイン IBM® SPSS® Modeler ツールバー追加して拡張子を使用できるようにします。

図 2-5
項目をメイン ツールバーに追加

この場合、動作を表すラベルを隠すことをお勧めします。

ノードのダイアログ ボックスまたは出力ウィンドウのツールバーに項目を追加することもできます。次の例のように、項目ラベルの表示または非表示を選択できます。

図 2-6
項目をダイアログ ボックスのツールバーに追加

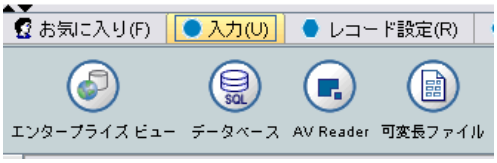


詳細は、 6 章 p. 153 ツールバー項目 を参照してください。

パレットおよびサブパレット

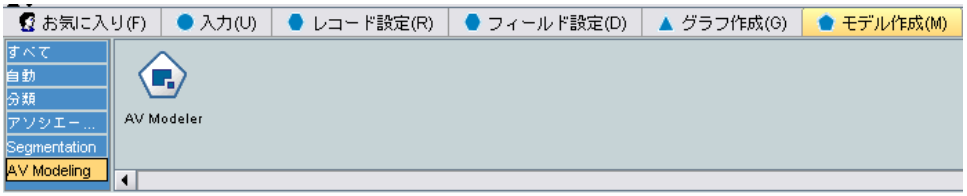
拡張子が新規ノードを定義する場合、任意の場所のノードを標準 IBM® SPSS® Modeler パレットまたはサブパレットのいずれかに配置できます。

図 2-7
標準パレットの新規ノード



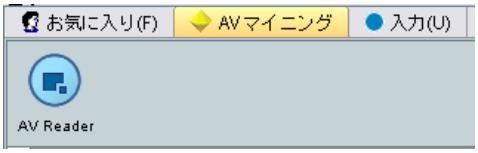
エントリを標準サブパレットに追加し、ノードをサブパレットからアクセスできるようにします。

図 2-8
新規ノードの標準パレットへのカスタム追加



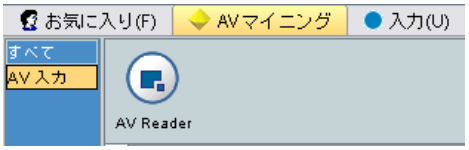
カスタム パレットを定義し、新規ノードを配置できます。

図 2-9
カスタムパレットの新規ノード



カスタム パレットにはカスタム サブパレットを配置できます。

図 2-10
カスタム パレットのカスタム サブパレットの新規ノード



詳細は、「ノード」(p. 66)および「User Interface (Palettes) セクション」(p. 59)を参照してください。

ノード アイコンのデザイン

CLEF で作成するそれぞれの新規ノードについて、画面上のノードを識別するアイコンの中央の画像を提供できます。

注 : IBM® SPSS® Modeler では指定がない場合に表示されるデフォルトを提供しているため、画像を提供する必要はありません（ノード開発の開始時に役に立ちます）。

図 2-11
CLEF アイコンのデフォルト イメージ

標準的な SPSS Modeler アイコンは、次の 3 つのレイヤーで作成されます。

- 枠線
- 背景
- 中央の画像

新規ノードの場合、中央の画像（**グリフ**）のみ提供する必要があります。SPSS Modeler では、枠線および背景の処理を行います。アイコンの背景レイヤーが見えにくいことのないよう、グリフ イメージの背景は透明である必要があります。このセクションでは、グリフの表示では色の付いた背景が透過度を示しています。

図 2-12
透過度を示す色の付いた背景のグリフ レイヤー

このようにして、通常の SPSS Modeler モデル作成アイコンが作成されます。

テーブル 2-3
ノード アイコンと生成されたモデル アイコンの構成

	ノード アイコン	生成されたモデル アイコン
枠線		None
背景		
グリフ		
表示される画像		

枠線(O)

ノードの機能は、アイコンの枠線の形状で識別されます。 詳細は、p. 10 ノードの概要 を参照してください。

ノードがキャッシュを有効化している場合、枠線の形状に縮小された文書のシンボルが追加されています。ノードに表示されている文書アイコンが白い場合、キャッシュが空であることを表しています。キャッシュがいっぱいになると、文書アイコンは濃い緑色になります。

テーブル 2-4
ノードの枠線およびキャッシュ ステータス

キャッシュ ステータス	例
キャッシュ なし	
有効なキャッシュ	
いっぱいになったキャッシュ	

さまざまな枠線のシンボルがシステムに提供され、IBM® SPSS® Modeler は必要な処理を行い、適切なときに適切なシンボルを表示します。

背景

生成されたモデルおよびモデル アプライヤ ノードのノード アイコン以外のノードアイコンの場合、背景の色が変わり、状態を表します。

テーブル 2-5
ノードの背景

State	色	例
選択解除	灰色	
Selected	青	

State	色	例
エラー	赤	
データベースで実行する動作	紫	

再度、背景の画像がシステムによって提供され、IBM® SPSS® Modeler で必要な処理を実行し、それぞれの状況に適切な背景を表示します。

グラフィックの必要条件

それぞれの新規 CLEF ノードに対し、次のバージョンのグリフ レイヤー画像を作成します。

- ストリーム領域のノードの場合、大きいサイズ（49 x 49 ピクセル）
- 画面下部のパレット マネージャ内のノードの場合、小さいサイズ（38 x 38 ピクセル）

メニューまたはツールバー、ブラウザまたは出力ウィンドウのタイトルバーにアイコンを表示する場合、次を作成する必要もあります。

- 縮小サイズ（16 x 16 ピクセル）

ノードがモデルを生成する場合、次を作成する必要もあります。

- 生成されたモデルのアイコン（金色のナゲット）にオーバーレイする場合、左下に移動したデザインの小さいサイズ（38 x 38 ピクセル）

注 : IBM® SPSS® Modeler に表示する場合、これらのサイズを超える画像は一部が切り取られます。

詳細は、 6 章 p.148 アイコン を参照してください。

カスタム画像の作成

ノードに作成する画像は、ノードの主要な機能を示す必要があります。海外のユーザーに対し、1 つの国に特定せず、他の国のユーザーに誤解されないような画像を使用する必要があります。

CLEF で使用する画像を作成する手順は、次のとおりです。

▶ 透過度をサポートするグラフィック パッケージを使用して、描画領域を適切なサイズに設定し、イメージ バージョンを描画します。

- それぞれのバージョン（大、小など）を、次の特性を持つ個々の .gif ファイルで保存します。
  - 透明な背景
  - 16 色（4 ビット）以上の色数

  画像の背景透過度の設定方法は、使用するグラフィック パッケージによって異なります。たとえば、背景色を透過度に直接設定できる場合、または透過色を指定し、画像の背景のこの色で「塗る」ことができる場合があります。

  画像ファイルの場合、IBM® SPSS® Modeler が最初に使用するファイルの名の表記方法に従うことをお勧めします。

テーブル 2-6
画像ファイル名の表記方法

画像の種類	ファイル名
Large	lg_node.gif
小(S)	sm_node.gif
縮小	node16.gif
生成されたモデル	sm_gm_node.gif

- 設定ファイルの画像ファイルを参照し（「画像ファイルをノード設定に追加」 を参照）、および新規ノードを SPSS Modeler に追加して（「CLEF 拡張のテスト」（ p.274 ）を参照）、画像の外観を検定します。

画像ファイルをノード設定に追加

画像ファイルを作成した場合、IBM® SPSS® Modeler を実行するコンピュータのフォルダに画像をコピーします。設定ファイルにおいて、SPSS Modeler インストール ディレクトリの ¥ext¥lib¥provider.nodename フォルダに関連する画像パスを宣言し、アクセスしやすいフォルダにファイルを展開する必要があります、 詳細は、 6 章 p.148 アイコン を参照してください。

設定ファイルで、大きなアイコン グラフィック ファイルおよび小さいアイコン グラフィック ファイルを、Node 設定内にある UserInterface セクションの Icons 要素によってカスタム ノードと関連させます。以下はその例です。

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

モデル ビルダーまたはドキュメント ビルダー ノードの場合、ModelOutput 設定（モデル ビルダー ノードの場合）または DocumentOutput 設定（ドキュメント ビルダー ノード）の UserInterface セクションの縮小（16 x 16 ピクセル）バージョンも参照します。 以下はその例です。

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

モデル アプライヤ ノードの場合、Node 設定の UserInterface セクションの生成されたモデルのバージョンも参照します。以下はその例です。

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

ダイアログ ボックスのデザイン

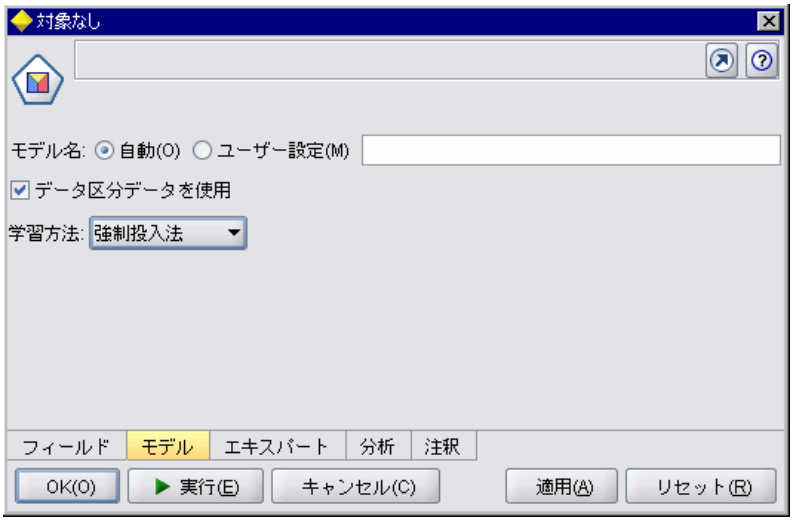
この項では、CLEF の一貫したダイアログ ボックスのデザインを支援する、標準 IBM® SPSS® Modeler ノード ダイアログ ボックスの特性を説明しています。

ノードのダイアログ ボックスについて

ノードのダイアログ ボックスは、エンド ユーザーが実行に関する設定の変更を可能にするインターフェイスを提供しています。ダイアログ ボックスでノードの動作を修正、変更できるため、ダイアログ ボックスの外観は非常に大切になります。このインターフェイスには必要なすべての情報を記載し、しかも使いやすくなければなりません。

ノードの動作は、ユーザーが対話処理できるユーザー インターフェイスの要素である、さまざまなダイアログ ボックス ベースの **コントロール** の使用によって変更されます。ダイアログ ボックスには、ラジオ ボタン、チェック ボックス、テキスト ボックス、およびメニューなど多くのコントロールで構成されています。CLEF では、ダイアログ ボックスにデザインできるさまざまなコントロールが提供されています。 詳細は、6 章 p. 167 プロパティ コントロール設定 を参照してください。

図 2-13
ラジオ ボタン、チェック ボックス、コンボ ボックス、およびタブを含むダイアログ ボックスの例



変更されたパラメータの種類によって、どのコントロールがダイアログ ボックスに表示されるかが決まります。種類によっては、代替コントロールがあるものもあります。設定ファイルの Tab 要素によって、新しいタブのオプションをグループ化できます。 詳細は、 p. 29 タブ領域 を参照してください。

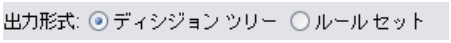
注 : 拡張子を実行する処理を指定していない場合でも、拡張子のユーザー インターフェイスの外観を検定できます。 詳細は、 10 章 p. 274 CLEF 拡張のテスト を参照してください。

ダイアログ ボックス デザインのガイドライン

ダイアログ ボックスのコントロールを定義する際は、次のガイドラインを参考にしてください。

- コントロールに表示ラベルがある場合、使用するテキストを慎重に考慮してください。テキストは、簡潔で正しく情報を伝達できるものでなければなりません。国際マーケット向けにデザインする場合、元のテキストとは大きく異なる翻訳されたテキストの長さについて考慮してください。
- パラメータに対して正しいコントロールを使用してください。たとえば、チェック ボックスは 2 つの値のみをとるパラメータに対して最善の選択とは限りません。IBM® SPSS® Modeler C5.0 ノードのダイアログ ボックスでは、ユーザーが出力タイプを選択できるラジオ ボタンを使用し、2 つの値[ディシジョン ツリー] または [ルール セット] のいずれかの値を持ちます。

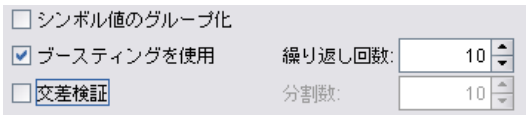
図 2-14
2 つの値間の選択



この設定は、[ディシジョン ツリー] というラベルを持つチェック ボックスとして表すことができます。このオプションが選択されると、出力タイプはディシジョン ツリーになります。選択が解除されると、出力はルール セットになります。結果は同じでも、ラジオ ボタンを使った方が、ユーザーにとってはわかりやすいものとなります。

- 通常ファイル名のコントロールは、ダイアログ ボックスの上部に配置するようにしてください。
- ノードのフォーカスを形成するコントロールは、ダイアログ ボックスの上の方に配置してください。たとえば、グラフ ノードはデータからフィールドを表示します。これらのフィールドを選択することがダイアログ ボックスの主な機能になるため、この場合はフィールド パラメータを上部に配置します。
- さらに詳細な情報が必要なため、ユーザーにオプションを選択させる場合、チェック ボックスやラジオ ボタンを使用します。たとえば、C5.0 ダイアログ ボックスで [ブースティングを使用] を選択する場合、分析に [繰り返し回数] を示す数字を入れる必要があります。

図 2-15
関連するコントロールを持つチェック ボックス



このような付加情報は、常にオプションの選択の後、右側か直下に配置します。

CLEF のダイアログ ボックスでは、標準的な SPSS Modeler のダイアログ ボックスと同じ方法で SPSS Modeler のコミット編集を使用します。[OK]、[適用]、またはターミナル ノードの場合は [実行] をクリックするまで、ダイアログ ボックスに表示される値はノードにコピーされません。同様に、ユーザーがキャンセルしてダイアログ ボックスを再表示するか、または [リフレッシュ] ボタンをクリックしない限り、ダイアログ ボックスに表示される情報が更新されることはありません（たとえば、現在のノードの上流における操作によりノードの入力フィールドが変更された場合）。

ダイアログ ボックス コンポーネント

ダイアログ ボックスには、次のコンポーネントがあります。

- タイトル バー

- アイコン領域
- 以下を組み込むツールバーおよびメニュー領域
  - ファイル、ノードの生成、表示、プレビュー、その他のボタン（ノードによって異なる）
  - 最大/標準サイズ ボタン
  - ヘルプ ボタン
- ステータス領域
- パネル領域
- タブ領域
- ボタン領域

それぞれのカスタム ノードには、ユーザーがノードを開いたときに表示されるダイアログ ボックスが必要です。設定ファイルに Tabs 要素を持つ UserInterface セクションを含む Node 要素がある場合、ノードを開くと前述のダイアログ ボックス コンポーネントがすべて表示されます。ノードの種類によって、タブ領域およびボタン領域の最小限の内容は次のとおりです。

テーブル 2-7
ノードの種類別のタブ領域およびボタン領域の最小限の内容

ノードの種類	タブ	ボタン
データ リーダー	[注釈]（ツールバー領域の［リフレッシュ］ボタンを含む）	[OK]、[キャンセル]、[適用]、[リセット]
データ変換	[注釈]	[OK]、[キャンセル]、[適用]、[リセット]
データ ライター	[公開]、[注釈]	[OK]、[キャンセル]、[実行]、[適用]、[リセット]
モデル ビルダー	[注釈]	[OK]、[キャンセル]、[実行]、[適用]、[リセット]

ノードの種類	タブ	ボタン
ドキュメント ビルダー	[注釈]	[OK]、[キャンセル]、[実行]、[適用]、[リセット]
モデル アプライヤ	[概要]、[注釈]	[OK]、[キャンセル]、[適用]、[リセット]

最初、ノードのダイアログ ボックスはユーザーがノードを開いた場合に、ノードのアイコンが表示するノードに付加されるように生成されます。ユーザーはダイアログ ボックスを移動できますが、新しい位置は次のノードを開くときには記憶されていません。ユーザーがダイアログ ボックスを移動し、別のダイアログボックスに一部または全体が隠れている場合、領域内の元のノードをダブルクリックすると、最初のダイアログ ボックスが再度前面に表示されます。ダイアログ ボックスはモードレス（同じユーザーの入力は常に同じ動作を実行）で、サイズ変更可能です。

ダイアログ ボックスのすべての編集可能なフィールドは、次のキーボードショートカットをサポートしています。

ショートカット	効果
Ctrl-C	コピー
Ctrl-V	貼り付け
Ctrl-X	切り取り

タイトル バー

ノードのダイアログ ボックスのタイトル バーでは、モデル名の前に IBM® SPSS® Modeler の金色のナゲットのアイコンの縮小バージョンが表示されます。テキストは、モデル名コントロールの設定から取得されます。デフォルトでは、右上に［閉じる］ボタン（X）が表示されます。

アイコン領域

ノード アイコンは、ダイアログ ボックスの左上部のアイコン領域に表示されます。

図 2-16
ノードのダイアログ ボックスの上部

これは、領域に表示される大きいバージョンではなく、メイン ウィンドウの下部のノード パレットにも使用されるアイコンの小さいサイズ（38 x 38 ピクセル）のバージョンです。

注： タイトル バーの左端に表示される縮小された金色のナゲットのアイコンは、すべてのノードのダイアログ ボックスにハードコード化されます。

ツールバーおよびメニュー領域

ダイアログ ボックスの最上部は、ツールバーおよびメニュー領域として使用されます。

データ　リーダーおよびデータ トランスフォーマー ノードのダイアログのこの領域には、入力データのサンプルを表示する［プレビュー］ボタンがあります。

データ リーダー ノードのダイアログには、ノードへの入力フィールドが変更された場合などにノードの表示される情報を更新する［リフレッシュ］ボタンがあります。

モデル アプライヤ ノードには［ファイル］、［ノードの生成］、［表示］のメニュー ボタンがあり、ユーザーはモデルのエクスポートまたはノードの新規作成など、さまざまな操作を実行することができます。モデル アプライヤ ノードには［プレビュー］ボタンもあり、この場合、ノード適用時に作成された追加の列とともに入力データのサンプルを表示します。

すべてのノードのダイアログ ボックス内の最上部領域の右側には、次の 2 つのボタンがあります。

- ［最大化/通常サイズ］ボタン
- ［ヘルプ］ボタン

［最大化/通常サイズ］ボタン

このボタンは、ダイアログ ボックスをフル スクリーン サイズにサイズを変更します。再度ボタンを押すと、ダイアログ ボックスを最大化する前のサイズに戻します。

[ヘルプ] ボタン

このボタンを使用すると、ノードのコンテキスト ヘルプを開きます。タブで分類されたダイアログ ボックスまたは出力ウィンドウの場合、該当するタブのヘルプが表示されます。F1 キーを使用して、ヘルプにアクセスすることもできます。

ステータス領域

ダイアログ ボックス丈夫の残りの領域は、情報、警告、エラー テキストの表示に使用されます。入力ノードは、ソース データ ファイルのフルパスおよびファイル名を表示します。個々のノードには、この領域に表示される他のノード固有の情報が含まれます。この領域に指定されたテキストは、2 行に制限されます。

パネル 領域

これはダイアログ ボックスの主要な領域で、すべてのコントロールが含まれ、ノードの領域を表示します。それぞれのタブには、異なるパネル領域があります。パネルの種類はそれぞれ、次のいずれかになります。

- テキスト ブラウザ
- 拡張オブジェクト
- Properties

新しいウィンドウで開き、パネルの操作ボタンで呼び出される個別のダイアログ ボックスであるサブパネルを指定することもできます。
詳細は、 6 章 p.159 パネル設定 を参照してください。

タブ領域

ノードのダイアログ ボックスには、次のタブが含まれています。

- ユーザーが提供したノード固有のタブ
- [要約] タブ（モデル出力オブジェクトおよびモデル アプライヤ ノードのみ）
- [注釈] タブ

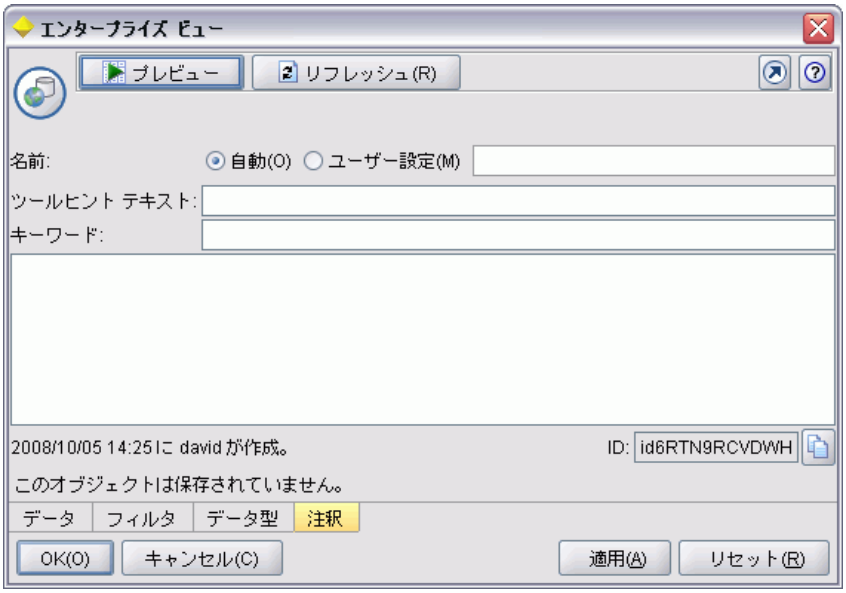
ノード固有のタブは、CLEF 設定ファイルの Tabs セクションで定義されます。 詳細は、 6 章 p.155 タブ を参照してください。

モデル出力オブジェクトおよびモデル アプライヤ ノードのダイアログ ボックスには、システムで提供された [要約] タブがあります。ここには、フィールド、構築設定、使用されたモデル推定プロセスなど、生成されたモデルについての情報が表示されます。結果は、特定の項目をクリックすると開いたり閉じたりできるツリーで表示されます。

図 2-17
[要約] タブ

[注釈] タブは、すべてのノードのダイアログ ボックスのシステムによって提供され、ユーザーはノードに関する情報を指定できます。この情報には、ノード名、ツールヒント テキスト、長いコメント フィールドが含まれます。

図 2-18
[注釈] タブ



名前 :デフォルトのノード名h、設定ファイルの Node 要素の Label 属性で指定されます（ノード p. 66 を参照）。ユーザーは、**カスタム** を選択し、[カスタム] 編集フィールドに名前を入力し、[**適用**] または [OK] をクリックして、ノードの名前を変更できます。新しい名前はセッション全体で保存されますが、デフォルト名は [**自動**] を選択して復元することができます。[注釈] タブで指定されたカスタム名は、ダイアログ ボックスの他のタブで指定されたカスタム名を上書きします。

ツールヒント テキスト :ここで指定されたテキストは、領域上のノードのツールヒントとして表示されます。ここでツールチップ テキストが指定されていない場合、ノード上にカーソルを移動してもツールヒントは表示されません。

キーワード :ユーザーはキーワードを指定して、プロジェクト レポートでおよび IBM® SPSS® Collaboration and Deployment Services Repository に保存されたオブジェクトを検索または追跡する場合に使用できます。

コメント パネル :この領域で、ユーザーはコメント テキストを入力できます。

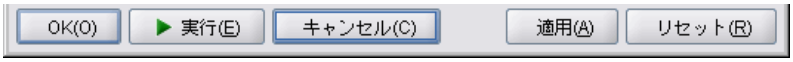
作成および保存情報 :これは、作成情報および名前、ファイルが保存された日時（日時の形式はロケールによって異なります）を表示する編集不可能なテキスト領域です。保存が行われていない場合、このフィールドには「この項目は保存されていません」と表示されます。

ボタン領域

すべてのダイアログ ボックスの下部に、[適用]、[リセット]、[OK]、および [キャンセル] が表示されます。ノードがターミナル ノード（ストリーム データを処理する実行可能なノード）の場合、[実行] ボタンも表示されます。

図 2-19
ダイアログ ボックスのボタン

図 2-20
ターミナル ノードのダイアログ ボックスのボタン



OK ：すべての設定を適用し、ダイアログ ボックスを閉じます。ダイアログ ボックスが最初にノードから開いた場合、このボタンにはフォーカス（ボタンの周りに青い四角形で表示）があり、また Enter キーを押しても［OK］の操作ができます。

キャンセル ：ダイアログ ボックスを閉じ、設定をダイアログ ボックスを開く前、または最後の［適用］操作以降のままにします。ダイアログ ボックス全体にフォーカスがある場合、Esc キーを押すと［キャンセル］の操作ができます。

実行 ：すべての設定を適用し、ダイアログ ボックスを閉じてターミナル ノードを実行します。

適用 ：ダイアログ ボックスの設定を保存し、下流の操作で使用できるようにします。

リセット ：ダイアログ ボックス全体をダイアログ ボックスを開いたときの値、または最後の［適用］操作以降の値にリセットします。

出力ウィンドウのデザイン

この項では、CLEF の一貫した出力ウィンドウのデザインを支援する標準 IBM® SPSS® Modeler 出力ウィンドウの特性を説明します。

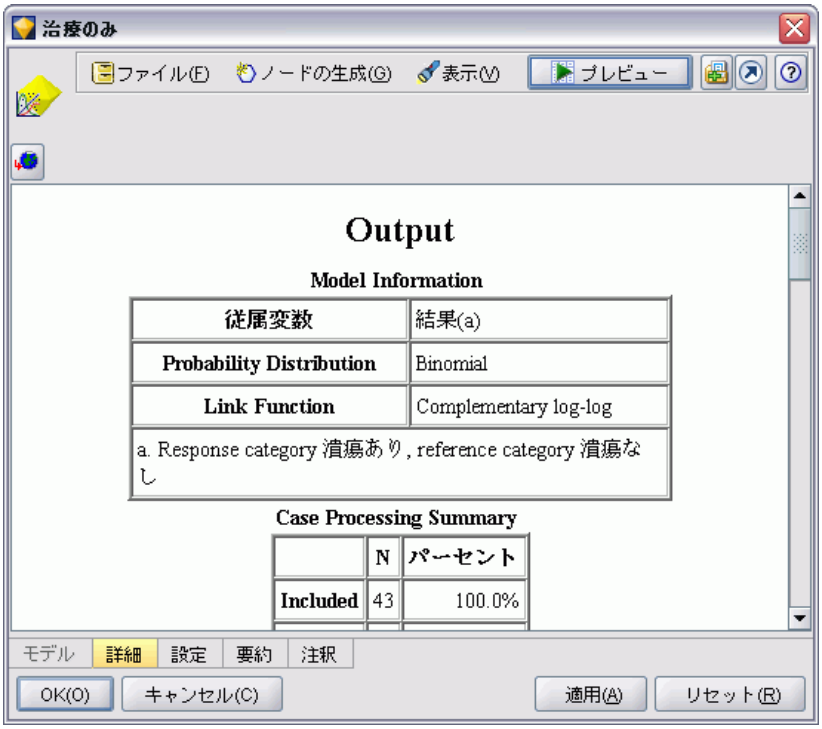
出力ウィンドウを使用すると、以下からのの出力を表示します。

- （モデルを適用する）データのセットの得点計算などのモデル
- グラフまたはレポートなどのドキュメント

詳細は、 6 章 p.142 ユーザー インターフェイスについて を参照してください。

通常の出力ウィンドウは、次のように表示されます。

図 2-21
モデル出力ウィンドウ



出力ウィンドウは、ノードのダイアログボックスと類似していますが、次の点が異なります。

- タイトル バーには、標準的な金色のナゲットアイコンではなく、ノード固有の縮小アイコンが表示されます。
- メイン ノード アイコンは省略されます。
- ツールバーおよびメニュー領域では、[最大化/通常] ボタンが省略されますが（ドキュメント出力ウィンドウでは、[閉じる] および [削除] ボタンと置き換えられる場合があります)、ウィンドウはマウスを使用するとサイズ変更できます。
- ステータス領域が省略されます。
- タブは通常、以下のとおりです。

  このオプションがモデル ノードで選択されている場合に予測値の重要度データを表示する（モデル出力ウィンドウの）[モデル] タブ

  出力の単一タブ

  モデルについて要約の詳細を表示する（モデル出力ウィンドウの）[要約] タブ

［注釈］タブ（注釈の値は出力を生成したノードから取得）

- ボタン領域には［OK］、［キャンセル］、［適用］および［リセット］ボタンがあります

CLEF では、前述のウィンドウに類似したデフォルトのモデル出力およびドキュメント出力ウィンドウが提供されます。通常、設定ファイルの ModelOutput または DocumentOutput 要素を使用する場合に表示されます。詳細は、 4 章 p. 66 オブジェクト識別子 を参照してください。

また、ModelOutput または DocumentOutput 要素を指定して、デフォルトの出力ウィンドウを独自にデザインしたカスタム ウィンドウを置き換えることができます。 詳細は、 6 章 p. 223 カスタム出力ウィンドウ を参照してください。

章 3

CLEF の例

サンプルについて

CLEF を理解するため、IBM® SPSS® Modeler インストールには、完全なソース コードとともにサンプル ノードのセットが含まれています。これらは、制限された機能を持つ基本的なノードで、CLEF の動作方法および使用方法を理解するために設計されています。これらのノードを今すぐに、またはいつでも使用できます。

サンプル ノードは次のとおりです。

- データ リーダー ノード (Apache Log Reader)
- データ変換ノード (URL Parser)
- ドキュメント ビルダー ノード (Web Status Report)
- モデル ビルダー ノード (Interaction)

使用する前に、ノードの例を有効化する必要があります。

サンプルの有効化

サンプルは、IBM® SPSS® Modeler インストールの一部として、Demos ディレクトリに圧縮された形式でインストールされています。次のようにファイルを適切な場所に抽出することでサンプルを有効化する必要があります。

SPSS Modeler がインストールされたコンピュータ上で次のように実行します。

1. SPSS Modeler が実行中の場合は終了します。
2. ファイル clef_examples_ext_lib.zip を、SPSS Modeler インストール ディレクトリの Demos フォルダに置きます。

© Copyright IBM Corporation 1994, 2012.

- clef_examples_ext_lib.zip の内容を、SPSS Modeler インストール ディレクトリの ¥ext¥lib フォルダに抽出します。

  SPSS Modeler のみのインストールの場合:

- clef_examples_ext_bin.zip の内容を、SPSS Modeler インストール ディレクトリの ¥ext¥bin フォルダに抽出します。

  SPSS Modeler および IBM® SPSS® Modeler Server の個別インストールによる設定の場合:

- clef_examples_ext_bin.zip の内容を、SPSS Modeler およびSPSS Modeler Serverインストール ディレクトリの両方にある ¥ext¥bin フォルダに抽出します。

- (UNIX サーバーのみ) 前のステップに加えて、clef_examples_ext_bin.zip で提供されるメイクファイルを使用して例のソース コードをコンパイルします。 詳細は、 p.40 ソース コードの検証 を参照してください。

  すべてのケースで、最後は次のようにします。

- SPSS Modeler を起動し、次のノードがノード パレットで表示されるようにします。

[パレット] タブ	ノード
入力	Apache Log Reader
フィールド設定	URL Parser
モデル作成	交互作用
出力	Web Status Report

データ リーダー ノード (Apache Log Reader)

データ リーダー ノードのサンプルは、Apache HTTP Web サーバーのアクセス ログ ファイルからデータを読み込む入力ノードです。アクセス ログには、Web サーバーが処理したすべての要求の詳細が含まれています。ログ レコードは、Combined Log Format と呼ばれる形式です。以下に例を示します。

```
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)"
```

サンプル ノードを使用して、ログ レコードを読みやすいテーブル形式に変換できます。

Apache Log Reader ノードを使用する手順は、次のとおりです。

- CLEF サンプルがまだ有効化されていない場合は、有効化します。 詳細は、 p.35 サンプルの有効化 を参照してください。
- IBM® SPSS® Modeler を開きます。
- ノード パレットの［入力］タブで、[Apache Log Reader] を選択してストリーム領域に追加します。
- ノードを編集します。［オプション］タブの［Apache ログ ファイル］フィールドで、次を入力します。

  demos_folder\combined_log_format.txt

  この場合、demos_folder は、IBM® SPSS® Modeler インストール ディレクトリの Demos フォルダです（$CLEO_DEMOS 形式は使用しないでください）。
- [OK] をクリックします。
- タイプ ノードをストリームに追加します。
- データ型ノードを編集します。[**値の読み込み**] をクリックしてデータを読み込み、[OK] をクリックします。
- テーブルノードをデータ型ノードに接続してストリームを実行します。ログ ファイルの内容は、テーブル形式で表示されます。
- 次の 2 つのサンプルと使用するためにストリームを保存します。

データ変換ノード（URL Parser）

データ変換ノード サンプルは、前述のサンプルで戻されたデータにより高度な処理を実行します。ID フィールド（各行に一意の値を表示）および URL が表示された入力フィールドを選択します。ノードでは、これら 2 つのフィールドで構成される出力データを生成し、生成された各フィールドに URL データを解析します。たとえば、次の URL レコードにクエリー文字列

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

が含まれている場合、レコードは次のように解析されます。

生成されたフィールド	内容
URLfield_server	http://www.dummydomain.co.uk
URLfield_path	/resource.php
URLfield_field	res_id
URLfield_value	89

URL Parser ノードを使用する手順は、次のとおりです。

- 前述のサンプルからのストリームが閉じられている場合は、開きます。ストリームには、Apache Log Reader とデータ型ノードが含まれています。
- ノード パレットの［フィールド設定］タブで、URL Parser ノードをデータ型ノードに接続します。
- URL Parser ノードを編集します。［ID フィールド］ドロップダウン リストで、[ReturnedContentSize] を選択します。［URL フィールド］ドロップダウン リストで、[ReferralURL] を選択します。[OK] をクリックします。
- テーブル ノードを URL Parser ノードに接続してストリームを実行します。ReturnedContentSize および ReferralURL フィールドは、次の 4つの生成されたフィールドへ追加で解析した ReferralURL と一緒に表示されます。ReferralURL_server、ReferralURL_path、ReferralURL_field および ReferralURL_valueです。

ドキュメント ビルダー ノード (Web Status Report)

ドキュメント ビルダー ノードのサンプルは、Web サーバー ログから渡されたデータを読み込み、HTML ファイルの形式でレポートを生成します。レポートは、さまざまな HTTP ステータス コード (200、302、404 など) を戻すログ レコードの割合を示すテーブルで構成されています。

Web Status Report ノードを使用する手順は、次のとおりです。

- 最初のサンプルからのストリームが閉じられている場合は、開きます。これは、Apache Log Reader とデータ型ノードが含まれているストリームです。ストリームに 2 番目のサンプルの URL Parser がある場合、そのノードはこのサンプルでは無視されます。
- ノード パレットの［出力］タブで、Web Status Report ノードをデータ型ノードに接続します。
- Web Status Report ノードを編集します。［ステータス コード フィールド］ドロップダウン リストで、[StatusCode] を選択します。[実行] をクリックします。出力ウィンドウで、レポートの内容が表示されます。

モデル ビルダー ノード (Interaction)

モデル ビルダー ノードのサンプルでは、他のサンプルとは独立して動作し、標準的（非対話式）な方法で単純なモデルを構築、または生成される前にモデルを対話処理できます。このモデルは、通信会社の顧客離れの予想を試みます。

Interaction ノードを使用する手順は、次のとおりです。

- CLEF サンプルがまだ有効化されていない場合は、有効化します。 詳細は、 p.35 サンプルの有効化 を参照してください。
- IBM® SPSS® Modeler の新規ストリームを作成します。
- Demos ディレクトリからファイル telco.sav をインポートする Statistics ファイル入力ノードを追加します。
- [データ型] タブで、[Read Values] をクリックした後、メッセージ ボックスの [OK] をクリックして確定します。
- [**解約**] フィールド（リスト内の最後のフィールド）の役割を [**対象**] に設定し、[OK] をクリックします。
- ノード パレットの [モデル] タブで、Interaction ノードを入力ノードに接続します。

標準（非対話式）モデル構築を検定する手順は、次のとおりです。

- ストリームを実行してストリーム内、そして画面右上の [モデル] パレットにモデル ナゲットを作成します。
- テーブルノードをモデル ナゲットに接続します。
- テーブル ノードを実行します。テーブル出力ウィンドウの右側にスクロールし、解約予測を表示します。フィールド **$I-churn** には予測値が表示されますが、**$IP-churn** では予測の確信度（0.0 ～ 0.1）が表示されます。

対話式モデル構築を検定する手順は、次のとおりです。

- Interaction モデル ビルダーのダイアログ ボックスの [モデル] タブで、[**インタラクティブ セッションを起動する**] を選択します。
- [**実行**] をクリックして、[対話検定] ダイアログ ボックスを表示します。
- [対話検定] ダイアログ ボックスで、[**構築タスクの開始**] をクリックしてモデル構築の進捗状況を表示します。
- モデル構築の操作が終了した場合、ダイアログ ボックスの [構築タスク] テーブルに追加された行を選択します。
- ダイアログ ボックス上部のツールバー領域で、黄色のダイアモンド型アイコンのボタンをクリックします。これにより、モデル出力オブジェクト（model_1）が画面右上の [モデル] パレットに生成されます。

対話的に生成されたモデルは、最初のモデルと同じですが、名前は異なります。[**構築タスクの開始**] からのプロセスを繰り返すと、model_2 などの同じモデルをさらに生成します。

設定ファイルの検証

CLEF の動作方法を理解するには、提供されたサンプルの設定ファイルを検証することが良い方法です。設定ファイルは、次の場所にあります。

install_dir\ext\lib\extension_folder\extension.xml

この場合、install_dir は IBM® SPSS® Modeler インストールディレクトリで、extension_folder が、次のいずれかになります。

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

¥ext¥lib の下に他の拡張フォルダが表示される場合がありますが、これらは、CLEF を使用して作成されたシステム提供の IBM® SPSS® Modeler ノードに関連します。これらのノードがインストールに含まれるかどうかは、ライセンス許可された SPSS Modeler モジュールによって異なります。設定ファイル全体で強調表示されている場合がありますが、**これらのファイルを決して変更しないでください**。変更した場合、これらのノードは正しく機能しない場合があります。その場合には、関連する SPSS Modeler 製品を再インストールする必要があります。システムに提供されたファイルへの変更は、IBM Corp. ではサポートされていません。

ソース コードの検証

参照用に、サンプル ノードの完全なソース コードも提供されています。すべてのサンプル ノードでは、C++ のサーバー側ライブラリを使用しますが、Interaction ノードのみ、クライアント側の Java クラスも使用します。

ソース コード ファイルは、サンプル ノードを有効化した場合に自動的に抽出され、次のようにインストールされます。

位置	内容
...¥ext¥lib¥spss.interaction¥src	親フォルダの ui.jar ファイル内の .class ファイルの Java ソース コード
...¥ext¥bin¥spss.apachelogreader¥src ...¥ext¥bin¥spss.interaction¥src ...¥ext¥bin¥spss.urlparser¥src ...¥ext¥bin¥spss.webstatusreport¥src	親フォルダ内の DLL の C++ ソースおよびプロジェクト ファイル

サンプルの削除

IBM® SPSS® Modeler のサンプル ノードを今後参照しない場合、次の手順で削除できます。

- IBM® SPSS® Modeler を終了します。
- SPSS Modeler インストールディレクトリ内の ¥ext¥bin および ¥ext¥lib ディレクトリからサンプル フォルダを削除します。標準 SPSS Modeler フォルダを誤って削除しないようにしてください。削除すると、関連する SPSS Modeler 製品を再インストールする必要があります。削除するフォルダは次のとおりです。
  - spss.apachelogreader
  - spss.urlparser
  - spss.webstatusreport
  - spss.interaction

  変更は、次回 SPSS Modeler を起動した場合に有効になります。

章
4

特性ファイル

特性ファイルの概要

すべての CLEF の拡張子には、すべての拡張子の特性を定義する XML ファイルが含まれている必要があります。このファイルは**特性ファイル**と呼ばれ、名前は常に extension.xml です。特性ファイルは、次の項で構成されています。

- **XML 宣言**:XML バージョンおよびそのほかの情報のオプションの宣言。
- **Extension 要素**:ファイルの主要部分で、後続の項をすべて含みます。
- **Extension Details セクション**:拡張子に関する基本情報を指定します。
- **Resources セクション**:リソース バンドル、JAR ファイル、共有ライブラリなど、拡張子が機能するために必要が外部リソースを指定します。
- **Common Objects セクション**:(オプション) モデル、ドキュメント、プロパティ タイプなど、拡張子の他のオブジェクトで使用または参照される項目を定義します。
- **User Interface (Palettes) セクション**:(オプション) ノードが表示されるカスタム パレットまたはサブパレットを定義します。
- **Object Definition セクション**:ノード、モデル出力、ドキュメント出力など、拡張子で定義されるオブジェクトを識別します。

各セクションには、(要素のコンポーネントなど) 静的な宣言、または (ノードの出力データモデルの計算など) シンプルな動的プロセスのいずれか、または両方が含まれています。CLEF 特性ファイルの全体の形式は、次のようになります。

```
<?xml version="1.0" encoding="UTF-8" ?>
<Extension ...>
  <ExtensionDetails .../>
  <リソース
    Resources セクション
  </Resources>
 <CommonObjects>
Common Objects セクション
  </CommonObjects>
 <UserInterface>
User Interface (Palettes) セクション
  </UserInterface>
  Object Definition セクション
   オブジェクトの定義
```

© Copyright IBM Corporation 1994, 2012.

```
    オブジェクトの定義
    オブジェクトの定義
  ...
</Extension>
```

コメント行

特性ファイルの任意のポイントで、次の形式のコメント行を含むことができます。

```
<!-- コメント テキスト -->
```

必須かオプションか

後続のセクションの要素の定義（通常**形式**の見出しで識別）では、「(必須)」と指定されていない限り、要素の属性および子要素はオプションです。要素のシンタックスの詳細は、A 付録 の「CLEF XML スキーマ」(p.280) を参照してください。

特性ファイルの例

ここに単純なデータ変換ノードの場合の CLEF 特性ファイルの完全な例を示します。

```
<?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>
```

ExtensionDetails 要素は、IBM® SPSS® Modeler によって内部で使用される拡張子についての基本情報を提供します。

Resources 要素は、ファイルで後で参照されるサーバー側のライブラリの場所を指定します。パスの指定は、ライブラリが \ext\bin\spss.urlparser\urlparser.dll の SPSS Modeler インストール ディレクトリにあることを示します。

この特性ファイルには CommonObjects 要素は含まれません。

Node 要素はノード自体のすべての情報を指定します。

- Properties では、ノードのダイアログのタブで後で使用する 2 つのプロパティが最初に宣言されます。
- UserInterface 要素は、この拡張子に固有のノード ダイアログのタブの外観およびレイアウトを定義します（他のタブは SPSS Modeler で提供されます）。
- Execution 要素は、ノード実行時に使用される項目を定義します。この場合、これらの項目はファイル内で最初に宣言されたサーバー側のライブラリで、実行によって特定のステータス コードを返す場合に表示される一連のメッセージです。
- OutputDataModel 要素は、このノードが実行するデータ変換を定義します。入力データ モデル（このノードに入力されるフィールドのセット）がここで定義されるフィールドのセットと置き換えられるよう定義され、出力データ モデル（モデルが継続して変更されない場合、ここより下流のすべてのノードに渡されるフィールドのセット）を構築し

ます。この例では、ノードは 2 つの元のフィールド（id_fieldname と url_fieldname）を変更しないまま通過しますが、url_fieldname から取得した名前の 4 つのフィールドを追加します。

この特性ファイルは、IBM® SPSS® Modeler インストールの一部として提供されたノードの例の 1 つから取得されます。 詳細は、 3 章 p.37 データ変換ノード（URL Parser）を参照してください。

XML 宣言

XML 宣言はオプションで、文字エンコードのフォーマットの詳細と共に、使用されている XML のバージョンを指定します。

例

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

Extension 要素

Extension 要素は、ファイルの主要な部分を構築し、その他のすべてのセクションが含まれます。形式は次のとおりです。

```
<Extension version="version_number" debug="true_false">
   Extension Details セクション
   Resources セクション
   Common Objects セクション
   User Interface (Palettes) セクション
   Object Definition セクション
</Extension>
```

ここでの意味は次の通りです。

version は拡張子のバージョン番号です。

debug はオプションです、true に設定されている場合、［**デバッグ**］タブをダイアログまたは CLEF ノードまたは出力に関連するフレームに追加し、オブジェクトに定義されたプロパティおよびコンテナにアクセスすることができます。デフォルト値は false です。 詳細は、 10 章 p.276［デバッグ］タブの使用 を参照してください。

Extension Details セクション

Extension Details セクションでは、拡張子に関する基本情報を提供します。

書式

```
<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"/>
```

ここでの意味は次の通りです。

providerTag（必須）は、拡張子のプロバイダを一意に識別する名前です。値は、文字列 spss を含むことはできません。内部での使用に保持されます。

id（必須）はこの拡張子を一意に識別する名前で、それに関するシステム メッセージに使用されます。規約により、拡張ファイルは IBM® SPSS® Modeler インストール ディレクトリの ¥ext¥lib¥providerTag.id というフォルダに置かれます。

label（必須）は、拡張子の表示名です。このテキストは、ノードが追加された場合にパレット マネージャの［名前］フィールドに表示されます。詳細は、 10 章 p.274 CLEF 拡張のテスト を参照してください。

version は拡張子のバージョン番号です。

provider は、この拡張子のプロバイダを識別します。このテキストは、ノードが追加された場合にパレット マネージャの［プロバイダ］フィールドに表示されます。デフォルトは、文字列 (unknown) です。

copyright は、拡張子の著作権の通知です。このテキストは、ノードが追加された場合にパレット マネージャの［著作権］フィールドに表示されます。

description は、拡張の目的の簡単な説明です。このテキストは、ノードが追加された場合にパレット マネージャの［説明］フィールドに表示されます。

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."/>
```

Resources セクション

このセクションは、拡張子が機能するために必要な外部リソースを定義します。

書式

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

ここでの意味は次の通りです。

Bundle は、一連のクライアント側のローカライズされたリソースを識別します。 詳細は、 p. 47 バンドル を参照してください。

JarFile は、クライアント側の Java jar ファイルを識別します。 詳細は、 p. 48 Jar ファイル を参照してください。

SharedLibraryは、サーバー側のライブラリまたは DLL を識別します。 詳細は、 p. 48 共有ライブラリ を参照してください。

HelpInfo は、拡張子のヘルプ情報があれば、その種類を指定します。 詳細は、 7 章 p. 226 ヘルプ システムの実装 を参照してください。

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>
```

バンドル

Bundle 要素は、.properties ファイルまたは Java .class ファイルのいずれかとして実装される、クライアント側のリソース バンドルを指定します (ローカライズのメッセージ テキストなど)。 詳細は、 8 章 p. 230 ローカライゼーション を参照してください。

書式

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

ここでの意味は次の通りです。

id（必須）は、このバンドルの一意の識別子です。

path（必須）は、この特性ファイルの親フォルダに関連するバンドル ファイルの場所を指定します。バンドルが .properties ファイルを参照する場合、パスには言語の拡張子または .properties 接尾辞が含まれる必要があります。

例

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

これは、リソース バンドルが特性ファイルと同じフォルダの messages.properties というファイルにあることを示します。

Jar ファイル

JarFile 要素は、この拡張の Java クラスや他のクライアント側のリソースを提供するクライアント側の Java アーカイブ（.jar）ファイルを指定します。

書式

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

ここでの意味は次の通りです。

id（必須）は、この .jar ファイルの一意の識別子です。

path（必須）は、.jar ファイルの親フォルダに関連するバンドル ファイルの場所を指定します。

例

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

これは、この拡張の .jar ファイルが特性ファイルと同じフォルダにあることを示します。

共有ライブラリ

SharedLibrary は、サーバー側の共有ライブラリまたは DLL を識別します。これは通常、ノード実行のサポートにのみ必要です。ライブラリが複数のモジュールを実装する場合、ノード設定の Execution セクションの Module 要素は、ライブラリ内の特定のモジュールを識別します。

書式

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

ここでの意味は次の通りです。

id（必須）は、この共有ライブラリの一意の識別子です。

path（必須）は、サーバー側のインストール ディレクトリの \ext\bin フォルダに関連する共有ライブラリの場所を指定します。パスは、共有ライブラリ ファイルの拡張子（.dll など）を含むことはできません。

例

次の共有ライブラリの宣言

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

は、共有ライブラリが次の場所からロードされることを指定します。

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

この場合、install_dir はサーバー側の CLEF コンポーネントがインストールされているディレクトリを示します。このライブラリは複数のモジュールを実装しているため、必要な特定のモジュール（supervisedBinning）は構築ノードの設定内の Module 要素によって識別され、次のライブラリの識別子を参照します。

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

ヘルプ情報

オプションの HelpInfo 要素は、この拡張に提供することができるヘルプの種類を示します。 詳細は、 7 章 p. 226 ヘルプ システムの実装 を参照してください。

Common Objects セクション

オプションの Common Objects セクションは、設定ファイルの別の場所で定義された要素間で共有できるオブジェクトを定義しますこのセクションのオブジェクト（プロパティの列挙など）の中には、必要な場所でローカルに定義される場合もあれば、（モデルやドキュメントなど）ここで定義される場合もあります。

書式

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

ここでの意味は次の通りです。

PropertyTypes を使用すると、共通のプロパティ定義をオブジェクト間で共有することができます。 詳細は、 p. 50 プロパティ タイプ を参照してください。

PropertySets は通常、ビルダー ノード、モデル出力オブジェクトおよびモデル アプライヤ ノードに同じセットのプロパティが含まれている場合に使用されます。 詳細は、 p. 52 プロパティ セット を参照してください。

ContainerTypes は、複雑なデータ構造をラップすることができるオブジェクトである、コンテナ タイプを定義します。 詳細は、 p. 53 コンテナ タイプ を参照してください。

Actions は、メニューまたはツールバーによってなど、ユーザーとの対話に関する基本情報を定義します。 詳細は、 p. 55 Actions を参照してください。

Catalogs は、サーバーが動的に生成する値のリストから 1 つまたは複数のオプションを選択できるコントロールを実装します。詳細は、 p. 56 カタログ を参照してください。

例

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

プロパティ タイプ

オプションの Property Types セクションによって、共通するプロパティ定義をオブジェクト間で共有することができます。これにより、たとえばプロパティの定義を複数の場所で複製するのではなく、1 つの場所に

表示することができるなど、メンテナンスしやすくなる場合があります。定義を共有すると、オブジェクトの新しいインスタンスが作成された場合に値がコピーされるさまざまなオブジェクトのプロパティ間の互換性を実現することもできます。

プロパティ タイプは、Common Objects セクションでのみ定義することができます。

書式

```
<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 の属性は次の通りです。

id（必須）は、プロパティ タイプの一意の識別子です。

isKeyed が true に設定されている場合、プロパティ タイプがキーであることを示します。キー プロパティは、ユーザー定義の制御によって、一連の操作をフィールドを関連付けます （「プロパティ コントロール」（p. 191 ）を参照）。isKeyed が true に設定されているときに、valueType 属性は structure に設定されていなければなりません。構造化プロパティの詳細は、「構造化プロパティ」（ p. 85 ）を参照してください。

isList は、プロパティが指定された値の種類の値のリスト（true）であるか、または単一の値（false）であるかを指定します。

max および min は、範囲の最大値および最小値を示します。

valueType は、次のいずれかになります。

- string
- encryptedString
- fieldName
- integer
- double
- boolean
- date

- enum（列挙型プロパティ p. 84 参照）
- structure（構造化プロパティ p. 85 参照）
- databaseConnection

Enumeration および Structure 子要素の属性は互いに排反します。Enumeration、Structure および DefaultValue 子要素は特定のケースで使用されます。「列挙型プロパティ」（ p. 84 ）、「構造化プロパティ」（ p. 85 ）、および「デフォルト値」（ p. 87 ）を参照してください。

プロパティ セット

Property sets は通常、ビルダー ノード、モデル出力オブジェクトおよびモデル アプライヤ ノードに同じセットのプロパティが含まれている場合に使用されます。たとえば、モデル ビルダー ノードはビルダーで設定できるプロパティのデフォルト セットを定義することができますが、実際はモデル アプリケーションまで使用されません。自動的に転送するために、モデル出力にも含まれる必要があります。

書式

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

id は、このプロパティ セットの一意の識別子です。

Property 要素の詳細は、「Properties」（ p. 71 ）を参照してください。

例

この例では、2 つのプロパティ（作成する予測の数、確率を含めるかどうか）の定義を示します。Common Objects セクションで、次のように定義します。

```
<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>
```

モデル ビルダー ノード、モデル出力オブジェクト、モデル アプライヤ ノードのそれぞれの定義に、次のような includePropertySets 属性が含まれます（モデル ビルダー ノードのみの定義を示します）。

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

コンテナ タイプ

コンテナは、モデルやドキュメントなどの複雑なデータ構造のプレースホルダとして動作するオブジェクトです。コンテナは特定の種類として定義され、コンテナ タイプはここで定義されます。次のコンテナ タイプを定義することができます。

- モデルの種類
- ドキュメント タイプ

コンテナ タイプはクライアントとサーバー間で転送、複製、ファイルまたはコンテンツ リポジトリに保存することができます。モデルは、モデル アプライヤ ノードがモデル出力オブジェクトから生成された場合に複製されます。

それぞれのコンテナ タイプには事前定義されたプロパティのセットが含まれますが、カスタム プロパティを追加することができます。コンテナ タイプは、Common Objects セクションでのみ定義することができます。

書式

Container Types セクションの形式は次のとおりです。

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

ここでの意味は次の通りです。

ModelType は、特定のモデル タイプの形式を指定します。 詳細は、p. 54 モデルの種類 を参照してください。

DocumentType は、特定のドキュメント タイプの形式を指定します。 詳細は、 p. 54 ドキュメント タイプ を参照してください。

例

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

モデルの種類

モデルは、アルゴリズム名、モデル タイプ、入力データ モデルおよび出力データ モデルなどの情報を提供する必要があります。モデル タイプの定義は、特定のモデル タイプの形式を指定します。

モデル タイプの情報は、特性ファイルのこの場所で静的に、またはモデルがモデル ビルダー ノードで構築されている場合は動的に指定されます。

書式

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

ここでの意味は次の通りです。

- id（必須）は、モデル タイプの一意の識別子です。
- format（必須）はモデルの形式を示し、utf8（テキスト）または binary のいずれかとなります。モデル形式は、静的情報の一部として指定する必要があります。

例

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

ドキュメント タイプ

document は、グラフまたはレポートなどの出力オブジェクトです。ドキュメント タイプの定義は、特定のドキュメント タイプの形式を指定します。

書式

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

ここでの意味は次の通りです。

- id（必須）は、ドキュメント タイプの一意の識別子です。
- format（必須）はドキュメント タイプの形式を示し、utf8（テキスト）または binary のいずれかとなります。

例

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

Actions

アクションは、メニューまたはツールバーによってなど、ユーザーとの対話に関する基本情報を定義します。各アクションは、ラベル、ツールヒントまたはアイコンなど、ユーザー インターフェイスにどのように表示されるかを定義します。一連のアクションは、アクションの各グループに定義されるクライアント側の Java クラスに処理されます。また、アクションは特定のオブジェクト内で定義されます。

書式

```
<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>
```

ここでの意味は次の通りです。

id（必須）は、このアクションの一意の識別子です。

label（必須）は、ユーザー インターフェイスに表示される、アクションの表示名です。

labelKey はローカライズ用のラベルを識別します。

description はアクションの説明です。たとえば、カスタム メニュー項目またはツールバーのアイコンのアクション ボタンなど、メニュー項目またはボタンのツールヒントのテキストとなります。

descriptionKey はローカライズ用の説明を識別します。

imagePath はアイコン イメージなど、グラフィック ファイルの場所です。特性ファイルがインストールされているディレクトリに関連した場所が指定されます。

imagePathKey は、ローカライズ用のイメージ パスを識別します。

mnemonic は、このコントロールを有効化するために Alt キーと組み合わせて使用するアルファベットの文字です（たとえば、値 S を指定すると、ユーザーは Alt + S キーを使用してこのコントロールを有効化できます）。

mnemonicKey は、ローカライズ用の mnemonic を識別します。mnemonic も mnemonicKey も使用されない場合、mnemonic はこのコントロールに使用できません。 詳細は、 6 章 p. 157 アクセス キーとキーボード ショートカット を参照してください。

shortcut は、このアクションを開始するのに使用できるキーボードのショートカットを示す文字列です（たとえば、CTRL+SHIFT+A）。

shortcutKey はローカライズ用のショートカットを識別します。shortcut も shortcutKey も使用されない場合、ショートカットはこのアクションに使用できません。 詳細は、 6 章 p. 157 アクセス キーとキーボード ショートカット を参照してください。

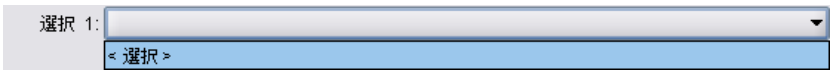
例

```
<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>
```

カタログ

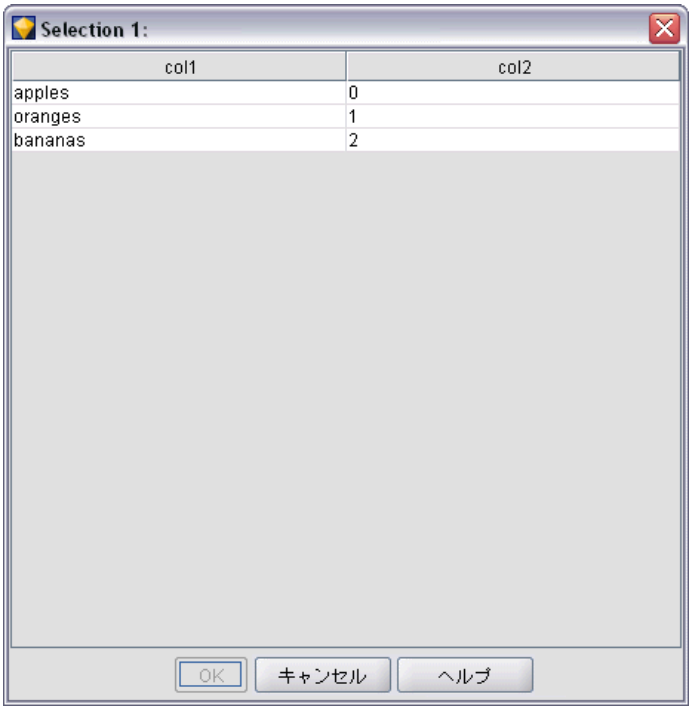
カタログを使用して、プロパティと、サーバーで動的に生成された値のリストから複数のオプションを選択できるコントロールを関連付けることができます。

図 4-1
値のリストから選択できるコントロール



<Select> エントリをクリックすると、値がコントロールにポップアップ リストで表示されます。

図 4-2
表示される値のリスト



ユーザーがリスト内の 1 行を選択すると、Catalog で指定された列の行値がコントロールに投入されます。

図 4-3
値のリストからの選択

書式

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

ここでの意味は次の通りです。

id（必須）は、このカタログの一意の識別子です。

valueColumn（必須）は、ユーザーが行を選択した場合にコントロールに投入される値を持つ列の数値です。列のナンバリングは 1 から始まります。

下の例のように、列ごとに Attribute 要素を、列の順に使用します。

ユーザーがカタログに関連するコントロールを有効にすると、getCatalogInformation 関数への呼び出しによって値リストを含むカタログがサーバーから取得されます。この関数は、値リストを含む XML ドキュメントを返します。 詳細は、 9 章 p. 244 ピア関数 を参照してください。

例

この例では、以前表示されたカタログ コントロールを定義されるために使用するコードの一部について説明します。3 つのカタログが定義され、ダイアログ タブの 3 つのさまざまな異なるコントロールと関連します。

まず、カタログは Common Objects セクションで定義されます。

```
<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>
```

次は、コントロールと関連するプロパティが、ノード定義の Properties セクションで定義されます。

```
<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>
```

ノード定義の User Interface セクションで、コントロールが定義され、プロパティへの参照によってカタログの定義と関連付けられます。

```
<UserInterface>
 <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>
```

User Interface (Palettes) セクション

これはオプションのセクションで、この拡張子でノードが表示されるカスタム パレットまたはサブパレットを定義する場合にのみ使用します。

拡張子がカスタム パレットまたはサブパレットを定義する場合、同じパレットまたはサブパレットに含まれるノードを定義する、後でロードされた拡張子は、User Interface (Palettes) を省略することができます。Node 要素にパレットを参照する customPalette 属性を含むだけでかまいません。拡張子は providerTag.id 値のアルファベット順でロードされます。ここでは、これらはこの拡張子の ExtensionDetails 要素の providerTag および id 属性の値です (「Extension Details セクション」(p. 45) を参照)。たとえば、拡張子 myco.abc は拡張子 myco.def の前にロードされます。

注 : User Interface (Palettes) セクションは、メインの User Interface セクションとは異なり、個々のオブジェクトの定義の一部として表示されます。詳細は、6 章、ユーザー インターフェイスの構築 p. 142 、

書式

User Interface (Palettes) セクションの形式は次のとおりです。

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

テーブル 4–1
パレットの属性

属性	説明
id	（必須）定義するパレットまたはサブパレットの一意の識別子。
systemPalette	システム パレットにサブパレットを追加する場合にのみ使用され、サブパレットが表示されるシステム パレットを識別します。 import – 入力 recordOp – レコード設定 fieldOp – フィールド設定 graph – グラフ作成 modeling – モデル作成（下記参照） dbModeling – データベース モデリング output – 出力 export – エクスポート
customPalette	カスタム パレットにサブパレットを追加する場合にのみ使用され、サブパレットが表示されるカスタム パレットを識別します。カスタム パレットを定義する Palette 要素の id 属性の値です。
relativePosition	カスタム パレットを定義する場合のみ使用され、画面下部のパレットのストリップの場所を指定します。 指定できる値は次のとおりです。 first last before after 値が before または after である場合、relativeTo 属性も必須です（下記参照）。 relativePosition が省略されている場合、パレットはストリップの最後に配置されます。
relativeTo	relativePosition の値が before または after の場合、relativeTo を使用して、このカスタム パレットが先行するまたは後続するパレットの識別子を指定します。パレットの識別子は、Node 要素のパレットの属性値として一覧表示されます（「ノード」（ p. 66 ）を参照）。
label	（必須）ユーザー インターフェイスに表示されるパレットまたはサブパレットの表示名。
labelKey	ローカライズ用のラベルを識別します。
description	ツールヒントのテキストは、カーソルがパレット タブの上にある場合に表示されます（サブパレットには使用されません）。また、この値はコントロールのアクセス可能な詳細な説明としても動作します。 詳細は、8 章 p. 238 アクセシビリティ を参照してください。

属性	説明
descriptionKey	ローカライズ用の説明を識別します。
imagePath	パレット タブで使用されるイメージの場所を識別します（サブパレットには使用されません）。特性ファイルがインストールされているディレクトリに関連した場所が指定されます。この属性を省略すると、イメージは使用されません。

例 – システム パレットへのノードの追加

会社がオーディオ データやビデオ データをマイニングする新しいアルゴリズムを開発し、アルゴリズムを IBM® SPSS® Modeler に統合したいと考えているとします。まずオーディオ ファイルおよびビデオ ファイルから入力を読み込むカスタム データ リーダー ノードを定義します。

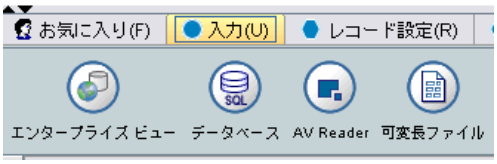
最初に、入力システムパレットに新しいデータ リーダー ノードを追加します。必要なことは、Node 要素の palette 属性によって入力パレットを識別することだけです。 詳細は、 p. 66 ノード を参照してください。

入力パレットのデータベース ノードの後にノードを追加するには、次を使用します。

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

これにより、次が作成されます。

図 4-4
標準パレットの新規ノード



例 – カスタム パレットの追加

標準 IBM® SPSS® Modeler パレットの使用は問題ありませんが、新しいノードにより高い重要度を与える必要があるとします。カスタム パレットを定義し、[お気に入り] パレットの後、[入力] パレットの前に配置します。まず、次のような、カスタム パレットを定義する User Interface (Palettes) セクションを追加する必要があります。

```
<UserInterface>
 <Palettes>
 <Palette id="AV_mining" label="AV Mining" relativePosition="before" relativeTo="import"
```

```
        description="Audio video mining"  />
    </Palettes>
</UserInterface>
```

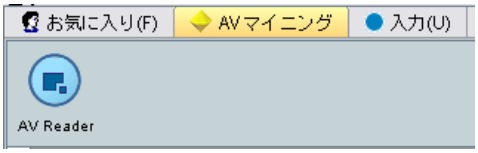
relativeTo 属性で、［入力］パレットの内部識別子 import を使用する必要があります。

その後、次のように Node 定義を変更します。

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

これにより、［お気に入り］システム パレットと［入力］システム パレットの間に［AV マイニング］パレットを配置します。

図 4–5
新しいパレットの新しいノード



例 – カスタム パレットへのカスタム サブパレットの追加

前述の例に続き、データ リーダー ノードを［AV マイニング］パレットの［AV ソース］サブパレットで動作させるとします。これをアーカイブするには、まず User Interface (Palettes) セクションに 2 番目の Palette 要素を追加してサブパレットを指定する必要があります。

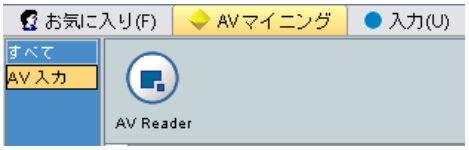
```
<UserInterface>
 <Palettes>
 <Palette id="AV_mining" label="AV Mining" description="Audio video mining" />
<Palette id="AV_mining.sources" customPalette="AV_mining" label="AV Sources" />
    </Palettes>
<UserInterface>
```

その後、サブパレットの識別子を参照する Node 要素を変更します。

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

［AV マイニング］タブをクリックすると、1 つは［すべて］、もう 1 つは［AV ソース］というラベルの付いた 2 つのサブパレットが表示されます。AV リーダー ノードがこれらのサブパレットの両方に表示されます。

図 4-6
新しいサブパレットの新しいノード



[AV マイニング] の追加の新しいサブパレットにもう 1 つ新しいノードを追加する場合、新しいノードは [すべて] サブパレットと新しいサブパレットの両方に表示されますが、[AV ソース] サブパレットには表示されません。

例 – システム サブ パレットへのノードの追加

オーディオおよびビデオ ソース データを処理するには、モデル ビルダー ノードを定義します。多くの標準的なサブパレットを含む標準［モデル作成］パレットにモデル ビルダー ノードを追加します。次のように指定して、［分類］サブパレットにモデル ビルダー ノードを追加し、ニューラル ネットワーク ノードの直前に配置します。

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

［分類］サブパレットが次のように表示されます。

図 4-7
標準サブパレットの新しいノード

ノードは、［モデル作成］パレットの［すべて］サブパレットの同じ関連位置にも追加されます。

例 – システム パレットへのカスタム サブパレットの追加

［分類］サブパレットのモデル ビルダー ノードの数について考えた場合、ユーザーは新しいノードに簡単に気が付かない場合があると考えました。ノードをより目立つようにする 1 つの方法は、［モデル作成］パレットに独自のサブパレットを追加し、そこにノードを配置することです。

そして次のように User Interface (Palettes) セクションをファイルに追加して、カスタム サブパレットを定義する必要があります。

```
<UserInterface>
 <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>
</UserInterface>
```

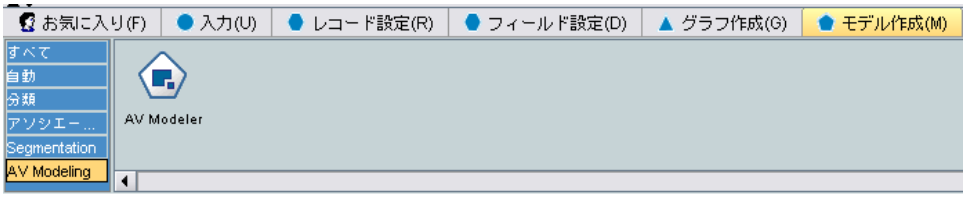
拡張するシステム パレットを識別するためには、明示的に **systemPalette** を指定する必要があります。

ノードのメインの User Interface セクションで、このサブパレットに表示されるよう指定します。

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

カスタム サブパレットは、常にシステム サブパレットの後にあり、次のようになります。

図 4-8
標準サブパレットへのカスタム サブパレットの新しいノードの追加



注 : [AV モデル作成] サブパレットにさらにノードを追加するには、AV Modeler 拡張子が最初にロードされている場合、設定ファイルに User Interface (Palettes) セクションは必要**ありません**。

カスタム パレットまたはサブパレットの非表示または削除

カスタム パレットまたはサブパレットを表示しない場合、IBM® SPSS® Modeler パレット マネージャを使用して、非表示または削除することができます。

非表示の操作は SPSS Modeler セッション全体で反映されますが、チェックボックスで制御するように元に戻すことができます。削除の操作は同じセッションで元に戻すことはできませんが、SPSS Modeler を再起動すると、特性ファイルまたは拡張子全体から削除しない限りは、再度表示されます。詳細は、 10 章 p. 279 CLEF 拡張のアンインストール を参照してください。

パレットを非表示または削除する手順は次のとおりです。

▶ メイン SPSS Modeler メニューから次の各項目を選択します。
ツール > パレット管理

▶ [パレット名] フィールドでパレットを選択し、次の作業を行います。
- パレットを非表示にするには、対応する [表示 ?] チェック ボックスのチェックを解除します。
- パレットを削除するには、[削除選択] ボタンをクリックします。

▶ [OK] をクリックします。

サブパレットを非表示または削除する手順は次のとおりです。

▶ メイン SPSS Modeler メニューから次の各項目を選択します。
ツール > パレット管理

▶ [パレット名] フィールドでパレットを選択し、次の作業を行います。

▶ [サブパレット] ボタンをクリックします。

▶ [サブパレット名] フィールドでサブパレットを選択し、次の作業を行います。
- サブパレットを非表示にするには、対応する [表示 ?] チェック ボックスのチェックを解除します。
- サブパレットを削除するには、[削除選択] ボタンをクリックします。

▶ [OK] をクリックします。

Object Definition セクション

この要素は、拡張子で最も目立つ部分です。Object Definition セクションは CLEF 特性ファイルの残りを構成し、拡張子のさまざまなオブジェクトを定義するために使用されます。次の種類のオブジェクトを定義することができます。

- nodes
- モデル出力オブジェクト
- ドキュメント出力オブジェクト
- インタラクティブ出力オブジェクト

ノードは、ストリームで表示されるオブジェクトです。**モデル出力オブジェクト**はモデル ビルダー ノードで生成され、メイン ウィンドウのマネージャ領域の [モデル] タブに表示されます。同様に、**ドキュメント出力オブジェクト**はドキュメント ビルダー ノードで生成され、同じ領域の [出力] タブに表示されます。**インタラクティブ出力オブジェクト**はインタラクティブ モデル ビルダー ノードで生成され、マネージャ領域の [出力] タブに表示されます。

Object Definition セクションは、3 つのオブジェクト定義の 1 つまたは複数で構成されています。

さまざまな種類のオブジェクトに定義することができる要素は、次のセクションで説明されています。これらの要素の中にはすべてのオブジェクト タイプに共通なものがあれば、ノードまたはモデル出力定義に固有のものもあります。オブジェクト固有の要素は、テキスト内でそのようなものとして示されています。

- オブジェクト識別子
- モデル ビルダー
- ドキュメント ビルダー
- プロパティ
- コンテナ
- ユーザー インターフェイス
- 実行
- 出力データモデル
- コンストラクタ

オブジェクト識別子

オブジェクト識別子は、オブジェクトの種類を示し、次のいずれかが使用されます。

<Node .../>

<ModelOutput .../>

<DocumentOutput .../>

<InteractiveModelBuilder .../>

また、オブジェクト識別子はスクリプトでオブジェクトがどのように公開されるかについての情報も提供します。scriptName 属性は、オブジェクトの一意の名前を示します。スクリプトでは、この属性を使用して特定のオブジェクトを指定することができます（ストリーム内のノード、[出力] タブの出力など）

ノード

ノード定義は、ストリームに表示されるオブジェクトを説明します。

書式

```
<Node id="identifier" type="node_type" palette="palette" customPalette="custom_palette"
        relativePosition="position" relativeTo="node" 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>
    <UserInterface>
     ...
</UserInterface>
    <Execution>
     ...
</Execution>
 <OutputDataModel ...>
...
    </OutputDataModel>
    <Constructors>
     ...
</Constructors>
</Node>
```

ノード定義内で許可された要素は、「Properties」 (p.71) で始まるセクションで説明されています。

テーブル 4-2
ノード属性

属性	説明
id	(必須) このノードの識別子。テキスト文字列。
type	(必須) ノードの種類。 dataReader - データを読み込むノード (入力パレット ノードなど)。 dataWriter - データを書き込むノード (エクスポート パレット ノードなど)。 dataTransformer - データを変換するノード (レコード/フィールド設定ノードなど)。 modelBuilder - モデル ビルダー ノード (モデル作成パレット ノード)。

属性	説明
	documentBuilder – グラフまたはレポートを作成するノード。 modelApplier – 生成されたモデルを含むノード ノード タイプは、パレットおよび領域上のノードのアイコンの形状を指定します。詳細は、 2 章 p. 10 ノードの概要 を参照してください。 ノード タイプが modelBuilder の場合、ノードの定義には ModelBuilder 要素が含まれている必要があります。「モデル ビルダー」 (p. 70) を参照してください。 ノード タイプが documentBuilder の場合、ノードの定義には DocumentBuilder 要素が含まれている必要があります。「ドキュメント ビルダー」 (p. 70) を参照してください。
palette	ノードが表示される標準 IBM® SPSS® Modeler パレットまたはサブパレットのいずれかの識別子は、次のとおりです。 import – 入力 recordOp – レコード設定 fieldOp – フィールド設定 graph – グラフ作成 modeling – モデル作成（下記参照） dbModeling – データベース モデリング output – 出力 export – エクスポート [モデル作成] パレットには、次のようにさまざまな標準サブパレットがあります。 modeling.classification – 分類 modeling.association –アソシエーション modeling.segmentation – セグメント化 modeling.auto – 自動化 palette 属性を省略すると、ノードは [フィールド設定] パレットに表示されます。 注 :palette 属性は、モデル ビルダー ノードにのみ使用されます。
customPalette	ノードが表示されるカスタム パレットまたはサブパレットの識別子。Palette 要素の id 属性の値で、ファイルのユーザー インターフェイス (パレット) セクションで指定されます。詳細は、 p. 59 User Interface (Palettes) セクション を参照してください。

属性	説明
relativePosition	パレット内のノードの位置を指定します。指定できる値は次のとおりです。 first last before after 値が before または after である場合、relativeTo 属性も必須です（下記参照）。 relativePosition が省略されている場合、ノードはパレットの最後に配置されます。
relativeTo	relativePosition の値が before または after の場合、relativeTo を使用して、このノードが先行するまたは後続するパレットのノードを指定します。relativeTo の値は、ノードのスクリプト名です。 標準 SPSS Modeler ノードの場合、スクリプト名は『SPSS Modeler スクリプトとオートメーション ガイド』の「プロパティ参照」で見つけることができますが、...node 接尾辞は使用しません（たとえば、データベース ノードの場合、databasenode ではなく database を使用します）。 CLEF ノードの場合、これはそのノードの scriptName 属性の値です。
label	（必須）パレット、領域、ダイアログに表示されるノードの表示名。
labelKey	ローカライズ用のラベルを識別します。
scriptName	スクリプトで参照された場合にノードを一意に識別する場合に使用します。 詳細は、 p. 103 スクリプトの CLEF ノードの使用 を参照してください。
helpLink	ヘルプ システムがあれば起動する場合に表示されるヘルプ トピックのオプション識別子です。識別子の形式は、ヘルプ システムの種類によって異なります（7 章、ヘルプ システムの追加 p. 226 を参照）。 HTML Help - ヘルプ トピックの URL JavaHelp - トピック ID
description	ノードのテキストによる説明
descriptionKey	ローカライズ用の説明を識別します。

ノード定義内で含まれる要素は、「モデル ビルダー」（ p. 70 ）で始まるセクションで説明されています。

例

ノード定義の例については、「特性ファイルの例」（ p. 43 ）を参照してください。

モデル出力

モデル出力定義は生成されたモデル（ストリームの実行の後マネージャ領域の［モデル］タブに表示されるオブジェクト）を説明します。

ファイルのこの部分のコード化の詳細は、「モデル出力」（ p. 119 ）を参照してください。

ドキュメント出力

ドキュメント出力定義は、ストリームの実行後マネージャ領域の［出力］タブに表示される生成されたテーブルまたはグラフなどのオブジェクトについて説明します。

ファイルのこの部分のコード化の詳細は、「ドキュメント出力」（ p. 136 ）を参照してください。

インタラクティブ モデル ビルダー

ファイルのこの部分のコード化の詳細は、「インタラクティブ モデルの構築」（ p. 120 ）を参照してください。

モデル ビルダー

この要素は、Node 要素の定義でのみ使用されます。

ファイルのこの部分のコード化の詳細は、5 章「モデルおよびドキュメントの構築」（ p. 107 ）を参照してください。

ドキュメント ビルダー

この要素は、Node 要素の定義でのみ使用されます。

ファイルのこの部分のコード化の詳細は、5 章「モデルおよびドキュメントの構築」（ p. 107 ）を参照してください。

モデル プロバイダ

この要素は、Node 要素の定義でのみ使用されます。

モデル出力オブジェクトおよびモデル アプライヤ ノードを定義する場合、ModelProvider 要素を使用して、モデルを保持するコンテナを指定することができます。モデルが PMML 形式で保存されるかどうかも指定することができます。PMML モデルは、カスタム ビューア、または ModelViewerPanel 要素で提供される標準 IBM® SPSS® Modeler モデル出力ビューアのいずれかを使用して表示することができます。 詳細は、 6 章 p. 166 モデル ビューア パネル を参照してください。

書式

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

ここでの意味は次の通りです。

container は、モデルを保持するコンテナの名前です。

isPMML は、モデルが PMML 形式で保存されるかどうかを指定します。

例

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

モデル アプライヤ ノードのコンテキスト内での ModelProvider の使用例は、「モデル ビューア パネル」(p. 166) の例を参照してください。

Properties

プロパティの定義は、名前と値のペアのセットで構成されています。数が多くなることが考えられるそれぞれのプロパティの定義は、1 つの Properties セクションに含まれています。

注 : プロパティが Properties セクション内で定義される場合、Properties セクションの定義が優先されるため、個々のプロパティ コントロールにプロパティを定義する必要はありません。このため、Properties セクション内でプロパティを定義することをお勧めします。

この規則の 1 つの例外は、label 属性に関連します。label 属性がプロパティ コントロールに定義されている場合、(label 定義だけでなく) プロパティ コントロールの宣言内にある任意のプロパティの定義は、Properties セクション内の対応する定義に対して優先されます。この例外はプロパティ コントロールのみに適用され、メニュー、メニュー項目、ツールバー項目などほかの種類のコントロールには適用されません。これらは直接的

に（メニュー）、または Action 要素を使用して間接的に（メニュー項目およびツールバー項目）、ラベルを明示的に定義する必要があります。

書式

```
<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>
```

Enumeration、Structure および DefaultValue 要素は特定のケースで使用されます。 詳細は、 p. 83 値の種類 を参照してください。

Property 要素の属性は次の通りです。

テーブル 4–3
Property 属性

属性	説明
name	（必須）プロパティの一意の名前。
scriptName	プロパティがスクリプト内で参照する名前。 詳細は、 p. 103 スクリプトの CLEF ノードの使用 を参照してください。
valueType	プロパティが取得することができる値の種類を指定します。次のいずれかを指定します。 string encryptedString fieldName integer double boolean date enum structure databaseConnection 詳細は、 p. 83 値の種類 を参照してください。
isList	プロパティが指定された値の種類の値のリスト（true）であるか、または単一の値（false）であるかを指定します。
defaultValue	このプロパティのデフォルト値です。単一の値の属性または複合要素として示され、指定された有効な値と一貫している必要があります。
label	ユーザー インターフェイスに表示される、プロパティの値の表示名。

属性	説明
labelKey	ローカライズ用のラベルを識別します。
description	プロパティの説明。
descriptionKey	ローカライズ用の説明を識別します。

プロパティは、オプションで有効な値がどのように指定されるのかを宣言します。

- 数値の場合、最小値または最大値となります。
- 文字列の場合、通常はフィールド選択となります（すべてのフィールド、すべての数値型フィールド、すべての離散型フィールドなど）がファイル選択となる場合もあります。
- 列挙する場合、有効な値のセットとなります。

また、キー プロパティは有効なキーがどのように指定されるかも宣言する必要があります。キー プロパティのキー タイプは文字列または列挙のいずれかです。 詳細は、 p. 50 プロパティ タイプ を参照してください。

プロパティに関連するオプションのデフォルト値は、関連するオブジェクトが作成される場合に評価されます。たとえば、ノードプロパティのデフォルトは、ノードの新しいインスタンスが作成されるごとに評価され、実行プロパティはノードが実行されるごとに評価されます。評価は、プロパティが宣言された順番に行われます。

プロパティ定義は、Common Objects セクションで宣言されたプロパティ タイプを参照する場合があります。

コンテナ

コンテナは、生成が Constructors セクションで定義される出力オブジェクトのプレースホルダです。

書式

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

ここでの意味は次の通りです。

name は、CreateModel または CreateDocument 要素の対象属性の値に対応し（「コンストラクタの使用」（ p. 137 ）を参照）、コンテナを Common Objects セクションで宣言されたコンテナ タイプのいずれかと関連付けます。

例

まず、コンテナ タイプは Common Objects セクションで宣言されます。モデルについてテキスト形式のコンテナが 1 つ、ドキュメント出力オブジェクトについて 2 つのコンテナ タイプ、1 つは HTML 出力のデフォルト（テキスト）形式、もう 1 つは zip 出力に対するバイナリ形式があります。

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

ノード定義の Execution セクションで、出力ファイルは Common Objects セクションで指定された識別子に対応するコンテナ タイプのコンテナ ファイルとして定義されます。

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

この後で、Constructors セクションは、ノード実行時に生成される出力オブジェクトを定義します。ここで、CreateModel 要素および CreateDocument 要素には Output Files セクションで指定されたコンテナ ファイルに対応する sourceFile 属性があります。

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

最後に Model Output セクションで、コンテナをモデル出力オブジェクトまたはドキュメント出力オブジェクトと関連付けます。Container 要素では、name 属性は指定した CreateModel 要素および CreateDocument 要素の target 属性に対応します。

```
<ModelOutput id="myoutput" label="My Model">
 <Containers>
 <Container name="model" />
       <Container name="advanced_output" />
       <Container name="zip_output" />
   </Containers>
    ...
</ModelOutput>
```

ユーザー インターフェイス

特性ファイルは、さまざまなユーザー インターフェイス コンポーネントをサポートし、オブジェクトを表示したり、コントロールやプロパティを変更することができます。コンポーネントのレイアウトやサイズ変更の操作、ほかのコントロールが変更された場合にコンポーネントを有効化するかまたは表示するかを指定する機能が提供されています。

User Interface セクションは、オブジェクトの外観を指定します。設定を使用して、ノード プロパティのダイアログまたは出力ウィンドウなど、基本のユーザー インターフェイス コンポーネントをカスタマイズすることができます。

User Interface セクションは、Node 要素設定の必要な部分です。

ファイルのこの部分のコード化の詳細は、6 章「ユーザー インターフェイスの構築」（ p. 142 ）を参照してください。

実行

この要素は、Node 要素の定義でのみ使用されます。

Execution セクションは、ノード実行時に使用されるプロパティおよびファイルを定義します。

書式

```
<Execution>
   <Properties>
    ...
</Properties>
 <InputFiles>
 <ContainerFile .../>
       ...
   </InputFiles>
   <OutputFiles>
      <ContainerFile .../>
```

```
        ...
    </OutputFiles>
    <Module ...>
        <StatusCodes .../>
    </Module>
 <Constructors .../>
</Execution>
```

Execution セクションには、ノードが実行されるごとに再度作成されるプロパティおよびノード実行中にのみ使用できるプロパティのセットの定義が含まれます。

実行の情報は、ノードの実行前に生成される入力ファイルのセット、実行中に生成される出力ファイルを定義することもできます。

いくつもの入力ファイルおよび出力ファイルを指定することができます。各入力ファイルは、ノードで定義されたコンテナと関連しています。各出力ファイルは、通常生成されたオブジェクトのコンテナを構築するために使用されます。入力ファイルまたは出力ファイルの形式は、Common Objects セクションのコンテナの宣言によって決まります。

例

Execution セクションの例については、「特性ファイルの例」（ p.43 ）を参照してください。

Properties（ランタイム）

このセクションは、ノード実行時にのみ使用できるランタイム プロパティのセットを定義します。

書式

形式は、要素定義の主要部分にある Properties セクションの定義に類似しています。 詳細は、 p.71 Properties を参照してください。

モデル ビルダー ノードまたはドキュメント ビルダー ノードの実行中、**サーバー一時ファイル** が作成され、モデル出力オブジェクトまたはドキュメント出力オブジェクトを保存します。サーバーは一時ファイルにアクセスし、オブジェクトをコンテナにラップされるクライアントに取り込みます。一時ファイルはここで指定する必要があります。

例

この例では、サーバー一時ファイルの指定方法を説明します。

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

入力ファイル

このセクションでは、ノードの実行前に生成される入力ファイルのセットを定義します。このコンテキストの入力ファイルは、サーバー上でのノード実行時に入力されるファイルです。たとえば、モデル アプライヤ ノードには、ノードの実行時に指定の入力ファイルに転送されるモデル コンテナがあります。

書式

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

入力ファイルの ContainerFile 要素の属性は次の表に示されているとおりです。

テーブル 4-4
コンテナ ファイルの属性 - 入力ファイル

属性	説明
id	コンテナ ファイルの一意の識別子。
path	入力ファイルを生成するサーバー上の場所（たとえば、サーバー一時ファイルの場所 - 「Properties（ランタイム）」（p. 76 ）を参照）。
container	サーバーに入力として送信されるオブジェクトを保持するコンテナの識別子。

例

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

出力ファイル

このセクションでは、サーバー上のノードの実行中に生成される出力ファイルを指定します。(モデル ビルダー ノードまたはドキュメント ビルダー ノードの結果など) 出力ファイルは、実行後にクライアントに転送されます。

書式

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

ContainerFile 要素の属性は次の表示示されているとおりです。

テーブル 4-5
コンテナ ファイルの属性 - 出力ファイル

属性	説明
id	コンテナ ファイルの一意の識別子。
path	クライアントに転送されるオブジェクトのサーバー上の場所 (たとえば、サーバー一時ファイルの場所 - 「Properties (ランタイム)」 (p.76) を参照)。
containerType	オブジェクトのコンテナ タイプ (モデル タイプまたはドキュメント タイプの ID など) の識別子。オブジェクトを適切な形式で転送することができます。 詳細は、p.53 コンテナ タイプ を参照してください。.

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>
```

モジュール

このセクションは、ノード実行中に使用されるサーバー側の共有ライブラリを指定します (たとえば、メモリーにロードされる DLL など)。

書式

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

ここでの意味は次の通りです。

libraryId は、Resources セクションの Shared Library 要素で宣言されるライブラリの識別子です。 詳細は、 p. 48 共有ライブラリ を参照してください。

name は、ライブラリが複数のノードで共有されている場合に使用され、実行される特定のノードを識別します。ライブラリが 1 つのノードにのみ使用される場合、name は空白にします。

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>
```

ステータス コード

ほとんどのプログラムは何らかのエラー チェックを実行してユーザーに必要なメッセージを表示します。通常、成功またはそのほかのステータスであることを示す整数を返します。サーバー側の API は、ノードを含むストリームの実行後にステータス コードを返すことができます。 詳細は、9 章 p. 264 状況詳細ドキュメント を参照してください。

Status Codes セクションでは、メッセージを特定のステータス コードと関連付け、ユーザーに表示することができます。

書式

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

ステータス コードの属性は、次の表のとおりです。

テーブル 4–6
ステータス コードの属性

属性	説明
code	メッセージが関連するステータス コード（整数値）。
status	次のような、ステータスの分類です。 success – ノードが正常に実行 warning – ノードの実行に警告 error – ノードの実行が失敗

属性	説明
message	ステータス コードが返される場合に表示されるメッセージ
messageKey	ローカライズ用のメッセージを識別します。

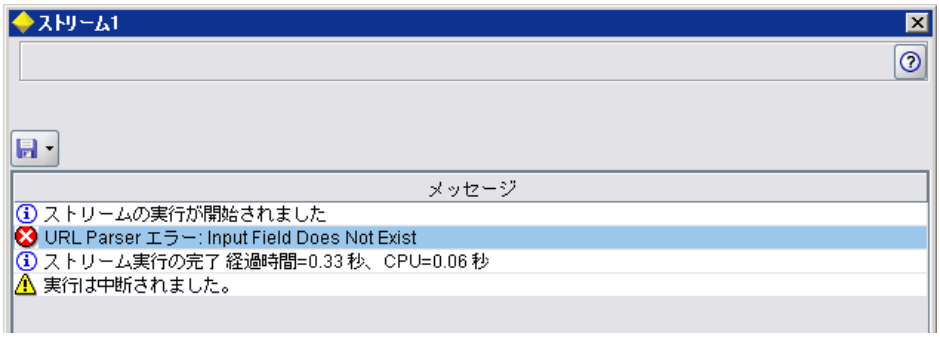
例

この例では、エラー メッセージのテキストに StatusCode 要素が含まれています。

```
<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>
```

実行時、サーバー側の API がステータス コード 3 を返した場合、ユーザーに次のメッセージが表示されます。

図 4-9
エラー メッセージの表示



次の例では、エラー メッセージのテキストは messageKey 属性に参照されます。

```
<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>
```

特性ファイルと同じフォルダにあるプロパティ ファイル（messages.properties など）には、実際のメッセージ テキストとその他の表示テキストが含まれています。

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

この方法は、ローカライズ用のすべてのテキストが 1 つのファイルにあるため、表示テキストを海外の市場向けにローカライズする必要がある場合に役立ちます。 詳細は、 8 章 p. 230 ローカライゼーション を参照してください。

出力データ モデル

この要素は、Node 要素の定義でのみ使用されます。

Output Data Model セクションは、データ モデルが特定のプロパティにどのように影響されるかを指定します。

出力データ モデルは、次の 3 つの方法のいずれかで指定することができます。

- 特性ファイルのフィールド セット定義機能を使用する。 詳細は、 p. 95 フィールド セット を参照してください。
- プロパティのセットおよび入力データ モデルを受け取り、データ モデルのインスタンスを返すデータ モデル プロバイダのインターフェイスを実装するクライアント側の Java クラスを使用する。
- プロパティのセットおよび入力データ モデルを受け取り、メタデータ ドキュメントを返す、サーバー側の共有ライブラリ コンポーネントを使用する。

Output Data Model セクションは、ノードのプロパティがノード全体のフィールドにどのように影響するかを定義します。出力データ モデルによって実行できるのは次のとおりです。

- 入力データ モデルを変更しないままにする
- 入力データ モデルを変更する
- 入力データ モデルを別のデータ モデルを置き換える

たとえば、入力ノードはプロパティ自体に影響を与えませんが並べ替え、フィールド作成ノードは新規フィールドを追加してデータ モデルを変更し、レコード集計ノードはデータ モデルを完全に置き換えます。

入力データ モデルを変更した場合、定義では新規フィールドを追加、または既存のフィールドを変更または削除することができます。データ モデルを置き換える場合は、新規フィールドのみ追加することができます。特

性ファイルは、入力フィールド セットまたは入力フィールド セットのフィールドのグループを示すキーまたはリストのプロパティ全体で反復する機能と共に、(データ型が入力フィールドに基づく新しいフィールドを作成する機能など) これらの基本操作をサポートしています。

書式

Output Data Model セクションの一般的な形式は次のとおりですが、これらのケースの特定の形式については、セクション 「複数フィールド設定コントロール」 (p. 186) および 「単一フィールド設定コントロール」 (p. 196) を参照してください。

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

出力データ モデルの属性は、次の表に示すとおりです。

テーブル 4-7
出力データ モデルの属性

属性	説明
mode	データ モデルへの影響。 extend - 新しいフィールドを既存のモデルに追加 fixed - 変更なし modify - 既存のフォールドを変更 (削除または名前を変更) replace - 既存のモデルを置換
libraryId	データ モデルを取得するサーバー側のコンテナの名前。

データ モデルの操作は、新しいフィールドを追加、または既存のフィールドを変更または削除する操作です。 詳細は、 p. 89 モータ モデルの操作 を参照してください。

例

OutputDataModel 要素が、特性ファイルの例に含まれています。 詳細は、 p. 43 特性ファイルの例 を参照してください。

Constructors

コンストラクタは、ストリーム内のノードの実行またはストリームへのオブジェクトの生成の結果として作成されるオブジェクトを定義します。

ファイルのこの部分のコード化の詳細は、「コンストラクタの使用」 (p. 137) で始まるセクションを参照してください。

共通機能

特性ファイルの複数のセクションで使用できる機能もあります。

- 値の種類
- 評価文字列
- 操作
- フィールドおよびフィールド メタデータ
- フィールド セット
- 役割
- 論理演算子
- 条件

値の種類

値の種類の宣言は、列、プロパティまたはプロパティ タイプがとることができる値の種類を指定します。

文字列および暗号化された文字列

valueType="string" の形式は、値がテキスト文字列であることを指定します。valueType="encryptedString" 宣言は、パスワードのフィールドなどユーザーが入力した場合に内容を非表示にする必要があるフィールドに関連するプロパティに使用されます。

フィールド名

値をフィールド名の形式から取得する必要がある場合、valueType="fieldName" の形式を使用します。

数学式、論理式、および日付式

値が数学的（整数または倍精度）、論理（true/false）または日付式である場合、valueType を integer、double、boolean または date にそれぞれ設定します。

列挙型プロパティ

列挙型プロパティは valueType="enum" 宣言の直後に続く Enumeration セクション内に含まれています。 詳細は、 p. 84 列挙型プロパティ を参照してください。

構造の宣言

valueType="structure" 宣言は、指定された他の属性を含む複合値を指定します。属性はプロパティと似ていますが、構造化またはキー化されません。詳細は、 p. 85 構造化プロパティ を参照してください。

- **キー指標 :** プロパティが単一の値であるか、またはテーブルの各値が指定された値の種類であるハッシュ テーブルであるかを指定します。
- **値セット :** 使用可能な値のセットがどのように決定されるかを指定します。
- **キー セット :** キー プロパティの場合、使用可能なキーのセットがどのように決定されるかを指定します。この情報は、使用する最も適切な種類のコントローラに関するヒントをユーザー インターフェイスに提供するためにも使用します。

データベース接続

user1@testdb など、ユーザーがデータベースにログオンすることができる接続文字列です。ログオンの詳細は、データベースに定義されている必要があります。 詳細は、 6 章 p. 183 データベース接続設定コントロール を参照してください。

列挙型プロパティ

列挙型プロパティは、値の事前定義されたリストから値を取得することができるプロパティです。

書式

列挙型プロパティでは、次のように値のリストが定義されている Enumeration セクションを使用します。

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

ここで PropertyType 属性は次のとおりです。

- id は、プロパティの種類の一意の識別子です。
- valueType は、プロパティ タイプが列挙型であることを示します。

ここで Enum 属性は次のとおりです。

- value（必須）は、値のリストに表示されるプロパティ値です。
- label（必須）は、ユーザー インターフェイスに表示される、プロパティ値の表示名です。
- labelKey はローカライズ用のラベルを識別します。
- description は、列挙された値の説明です。
- descriptionKey はローカライズ用の説明を識別します。

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>
```

構造化プロパティ

構造化プロパティは、ダイアログのテーブル コントロールなど、グリッドのような構造で使用されるプロパティです。

書式

構造化プロパティの形式では、次のように構造が定義されている Structure セクションを使用し、多くの Attribute 要素で構成されています。

```
<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>
```

ここで PropertyType 属性は次のとおりです。

- id は、プロパティの種類の一意の識別子です。

- valueType は、プロパティ タイプが構造型であることを示します。
- isList は、プロパティが指定された値の種類の値のリスト（true）であるか、または単一の値（false）であるかを指定します。

ここで Attribute 属性は次のとおりです。

- name（必須）は、列の識別子です。
- valueType は、列の内容がとることができる値の種類を指定します。次のいずれかを指定します。

  string

  encryptedString

  integer

  double

  boolean

  date

  enum
- isList は、プロパティが指定された値の種類の値のリスト（true）であるか、または単一の値（false）であるかを指定します。このようにして、キー プロパティを既知の属性の固定セット（特定のフィールドで実行されるさまざまなデータ集計操作を示す Boolean 属性など）または値のリスト（その他の同じフィールド名と関連するフィールド名のリストなど）と関連付けることができます。
- label（必須）は、ユーザー インターフェイスに表示される、アクションの表示名です。
- labelKey はローカライズ用のラベルを識別します。
- defaultValue は、表示時に列に表示される値です。
- description は、列の説明です。
- descriptionKey はローカライズ用の説明を識別します。

例 – テーブル コントロール

テーブル コントロールでの構造化プロパティの使用方法については、「テーブル コントロール」（ p. 200 ）を参照してください。

例 – キー プロパティ タイプ

最初の例では、関連する各値が固定されたセットの操作のフィールドに適用する集計操作を示す構造であるキー プロパティ タイプを示します。

```
<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>
```

aggregateOps プロパティ タイプを使用するよう宣言したプロパティは、次のようになります。

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

ここで、プロパティは異なるキーを持つ複数の値で構成されています。たとえば、キー name はフィールドの名前です（MIN、MAX など）。

キー プロパティ タイプの次の例では、関連する各値は単一の属性を含む構造です。この場合、属性はフィールドに適用される乗数を示す倍精度の式の一覧です。

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

multiplierOps プロパティ タイプを使用するよう宣言したプロパティは、次のようになります。

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

デフォルト値

DefaultValue 要素を使用して、サーバー一時ディレクトリ、ファイル、またはその両方を指定します。モデル出力オブジェクトまたはドキュメント出力オブジェクトを保存するために作成されます。

書式

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

ここで basename（必須）は、一時ディレクトリまたはファイルの名前です。

例

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

評価文字列

特性ファイルで宣言された文字列の中には、プロパティ名の参照が含まれるものもあります。これらの文字列は、評価文字列とも呼ばれます。

プロパティ参照の構文は、次のとおりです。

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

評価文字列にアクセスした場合、プロパティ参照は参照されたプロパティの値と置き換えられます。プロパティが存在しない場合、エラーが発生します。たとえば、新規フィールドを追加する場合、ノード定義に my_new_field というプロパティ、そしてこのプロパティの値を編集することができる User Interface セクションのコントロールが含まれている場合があります。

例

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

操作

特性ファイルの特定のセクションは、フィールドの追加、コンポーネントの作成、プロパティの初期化などの操作をサポートしています。操作をサポートするセクションは次のとおりです。

- 出力データ モデル（入力ノードおよびプロセス ノード）
- 入力データ モデルおよび出力データ モデル（コンポーネント）
- 出力オブジェクト作成（モデル ビルダー ノードおよびドキュメント ビルダー ノード）
- モデル アプライヤ作成（モデル出力）

操作は、次のタイプに分けられます。

- データ モデル 操作： AddField, ChangeField, RemoveField
- 反復： ForEach

モータ モデルの操作

データ モデルで実行することができる操作は次のとおりです。

- 新規フィールドを既存のデータ モデルに追加
- データ モデルの既存のフィールドを変更
- データ モデルのフィールドを削除

フィールドの追加

AddField 要素によって、新規フィールドを既存のデータ モデルに追加することができます。

書式

```
<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="役割" tag="propensity_type" value="value" >
   <Range min="min_value" max="max_value" />
</AddField>
```

AddField の属性は、次のとおりです。

テーブル 4-8
AddField 属性

属性	説明
prefix	モデル出力フィールドを示すなど、フィールド名に追加される接頭辞。
name	（必須）追加されるフィールドの名前。ハードコード化された文字列（field8 など）またはフィールドを参照する評価された文字列（${target} など）のいずれかです。 詳細は、 p. 88 評価文字列 を参照してください。
direction	フィールドが入力となるか対象となるかなど、フィールドの役割。in、out、both、partition または none のいずれか。 詳細は、 4 章 フィールドの役割の設定 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。
directionRef	フィールドの方向がフィールドを参照する評価された文字列（${field1} など）で識別されるフィールドの方向から取得されるよう指定します。 詳細は、 p. 88 評価文字列 を参照してください。
fieldRef	フィールドを参照する評価された文字列（${field1} など）で識別されるフィールドの対応する値からすべての参照値（directionRef、missingValuesRef、storageRef および typeRef）を取得するよう指定します。 詳細は、 p. 88 評価文字列 を参照してください。
group	フィールドがフィールド グループのメンバーとなるよう指定します。 詳細は、 5 章 p. 116 モデル フィールド を参照してください。
label	追加されるフィールドのラベル。

属性	説明
missingVal-uesRef	フィールドを参照する評価された文字列（${field1} など）で識別されるフィールドの欠損値特性から取得するよう、欠損値を処理する方法を指定します。 詳細は、 p. 88 評価文字列 を参照してください。
storage	フィールド値のデータ ストレージ タイプ - integer、real、string、date、time、timestamp または unknown のいずれか。
storageRef	ストレージ タイプがフィールドを参照する評価された文字列（${field1} など）で識別されるフィールドのストレージ タイプから取得されるよう指定します。 詳細は、 p. 88 評価文字列 を参照してください。
targetField	モデル出力フィールドについて、この新しいフィールドのデータが取得される対象フィールドを指定します。ハードコード化された文字列（field8 など）またはフィールドを参照する評価された文字列（${target} など）のいずれかです。 詳細は、 p. 88 評価文字列 を参照してください。
type	フィールド値のデータ型 - auto、range、discrete、set、orderedSet、flag または typeless のいずれか。 詳細は、 4 章 尺度 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。
typeRef	データ型がフィールドを参照する評価された文字列（${field1} など）で識別されるフィールドのデータ型から取得されるよう指定します。 詳細は、 p. 88 評価文字列 を参照してください。
role	モデル出力フィールドにあるデータの種類 - unknown、predictedValue、predictedDisplayValue、probability、residual、standardError、entityId、entityAffinity、upperConfidenceLimit、lowerConfidenceLimit、propensity、value または supplementary のいずれか。 詳細は、 p. 97 役割 を参照してください。
tag	role の値が propensity の場合のみ使用。傾向タイプを示し、RAW または ADJUSTED となります。
value	新規フィールドが含む値がフィールドを参照する評価された文字列（${field1} など）で識別されるフィールドから取得されるよう指定します。詳細は、 p. 88 評価文字列 を参照してください。

Range の属性は、次のとおりです。

テーブル 4-9
範囲の属性

属性	説明
min	フィールドが受け入れることができる最小値。
max	フィールドが受け入れることができる最大値。

例

次の例では、field8 という文字列を追加します。

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

次の例では、フィールド追加時にどのように参照をプロパティ名に追加することができるかを示します。ここで、フィールドは以前定義されたプロパティ prop1 の値に一致する名前で追加されます。

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

次の例では、対象フィールドの名前が field1 の場合、モデルは $S-field1 という出力フィールドを作成し、field1 の予測された値を保持します。

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

次の例では、0.0 ～ 1.0 の確率スコアを保持するモデル出力フィールドを追加します。

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

最後の例では、各モデル出力フィールドについて、0.0 ～ 1.0 の確率スコアを保持する出力フィールドが追加され、変数 fieldValue の値から値を取得します。

```
<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>
```

詳細は、 p. 88 評価文字列 を参照してください。

フィールドの変更

ChangeField 要素によって、データ モデルの既存のフィールドを変更することができます。

書式

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

ChangeField の属性は、次のとおりです。

テーブル 4-10
ChangeField 属性

属性	説明
name	(必須) 変更されるされるフィールドの名前。
fieldRef	フィールドの参照値。
direction	フィールドが入力となるか対象となるかなど、フィールドの役割。in、out、both、partition または none のいずれか。 詳細は、 4 章 フィールドの役割の設定 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。
storage	フィールド値のデータ ストレージ タイプ - integer、real、string、date、time、timestamp または unknown のいずれか。
type	フィールド値のデータ型 - auto、range、discrete、set、orderedSet、flag または typeless のいずれか。 詳細は、 4 章 尺度 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

Range の属性は、次のとおりです。

テーブル 4-11
範囲の属性

属性	説明
min	フィールドが受け入れることができる最小値。
max	フィールドが受け入れることができる最大値。

フィールドの削除

RemoveField 要素によって、データ モデルのフィールドを削除することができます。

書式

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

fieldRef はフィールドの参照値です。

ForEach 要素による反復

同じ操作を繰り返し実行してそれぞれの値のセットを処理すると役に立つ場合があります。特性ファイルでは、順番に提供されたセットの各値に一時プロパティを結びつける単純な ForEach の反復をサポートしています。ForEach ループを設定して、次のいずれかの方法で反復することができます。

- オプションのステップ サイズで 2 つの整数間
- リスト プロパティの値全体

- キー プロパティのキー全体
- フィールド グループのフィールド全体

書式

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

ここでの意味は次の通りです。

var（必須）は、反復が適用される値を含むフィールドを指定します。

from および to は、オプションの属性 step で整数のステップ サイズを指示して、反復の下限および上限を示す整数（または整数を計算する式）を指定します。

inFields, inFieldValues と inProperty は from/to/step 形式に対する代替です。

- inFields は、次の中から反復を実行するフィールド セットを指定します。
  inputs – ノードの入力フィールド
  outputs – ノードの出力フィールド
  modelInput – モデル署名で指定された入力フィールド
  modelOutput – モデル署名で指定された入力フィールド
- inFieldValues は、フィールド名（またはフィールド名を示すプロパティ）およびフィールドのメタデータの値の反復を指定します。」
- inProperty は、反復を実行するプロパティの名前を指定します。

ForEach 要素内で指定できるデータ モデル操作は、AddField 要素、ChangeField 要素または RemoveField 要素のいずれかです。 詳細は、p. 89 モータ モデルの操作 を参照してください。ForEach 要素は入れ子にすることもできます。

例

次は、操作を 10 回実行します。

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

次は、整数プロパティで指定された回数だけ、操作を実行します。

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

```
</ForEach>
```

次の例で、処理はノードの出力フィールドの値だけ反復します。

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

次の例では、**${field}** で識別されたフィールドのメタデータの値だけ反復します。

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

次の例では、リスト プロパティの値だけ反復します。

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

次の例では、キー プロパティのキー値の分反復します。

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

フィールドおよびフィールド メタデータ

ノード、モデル、およびデータ ソースは**データ モデル プロバイダ**として動作します。これらは他のオブジェクトからアクセスできるフィールド メタデータを定義することができます。

データ モデル プロバイダには、入力データ モデルおよび出力データ モデルが含まれます。たとえば、フィールドを追加して入力モデルを超える場合や既存のモデルを変更する場合など、出力データ モデルは入力データ モデルについて定義することができます。

これらのオブジェクトはそれぞれ、要件が異なります。

ノード : 入力データ モデルを参照することができますが、変更することはできません。出力データ モデルは入力データ モデルに基づいたり、置き換えたりすることができます。ノード プロパティまたは入力データ モデルが変更される場合は、出力データ モデルを再度計算します。モデル アプライヤ ノードの出力データ モデルは、モデル コンポーネントの出力データ モデルを参照することもできます。

モデル： デフォルトでは、入力データ モデルおよび出力データ モデル（モデル署名）は、モデルの作成時に使用された入力フィールドおよび出力フィールドの設定に基づきます。モデル構築プロセスは、必要な入力フィールドおよび生成された出力フィールドを定義するメタデータ ファイルを返すことが理想的です。モデル署名をいったん定義すると、変更することはできません。ただし、モデル アプライヤ ノードのプロパティはアプライヤ ノードのデータ モデル出力を変更することができます。たとえば、これらのプロパティはクラスタ ID が文字列として返されるのか整数として返されるのか、生成される一連の ID の数を定義します。また、モデル署名は通常、出力を「"out"」のフィールドの役割（direction）を持つものとして指定し、ノードは多くの場合役割が「"in"」の出力を生成します。

データ ソース： データ リーダー ノードで使用されるデータ ソースは出力データ モデルを指定することができます。入力データ モデルは常に空です。

フィールド セット

フィールド セットを多くの場所で使用して、データ モデル プロバイダのフィールドのサブセットを選択することができます。データ モデル プロバイダは、囲むオブジェクトまたは囲むオブジェクトのコンテナの場合があります。フィールド フィルタの最初の状態は、使用可能なすべてのフィールドを含めて特定の種類のフィールドを除外するか、空のフィールドのセットで開始し、必要なフィールドを含めたり新規フィールドを追加する場合があります。

次の例では、拡張ノードがどのように出力データ モデルを指定することができるかを示します。キー フィールドは、keys というリスト プロパティで指定され、名前のプロパティで指定される生成可能なオプションのレコード度数フィールドが後に続きます。

```
<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>
```

フィールドセットおよびモデル構築

次の例では、モデル アプライヤがどのようにして以前作成されたモデル コンポーネントの情報を使用して出力フィールドを生成できるかを示します。

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

AddField および ForEach は、使用される入力データ モデルまたは出力データ モデルを指定すると共にデータ モデル プロバイダを指定します。データ モデル プロバイダのフィールドのセット（またはサブセット）を指定するメカニズムを提供します。デフォルトのプロバイダは、デフォルトで使用される入力フィールド セットを持つ囲み要素（たとえば、作成されるオブジェクト以外）を示す this です。指定されるフィールド セットがない場合、使用可能なフィールドはすべて使用されます。

フィールド セットはストレージ、データ型、フィールドの役割または名前に基づきます。名前に基づく場合、リスト プロパティへの参照が必要です。フィールド セットは完全（デフォルト）または空の場合があります。完全の場合はフィールドを除外することができ、空の場合はフィールドを含むことができます。複数の値を個々のフィルタそれぞれに指定することができ、これらの値は「交差」または「and」の演算子として動作します。例を次に示します。

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

（include="none" で指定された）空のフィールド セットで開始し、フィールドの役割（direction）が「"in"」の文字列ストレージのフィールドが含まれます。

次にもう 1 つの例を示します。

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

ここでは、使用可能なすべてのフィールド（include="all" 属性で指定され、デフォルトの動作）が含まれ、typeless のデータ型は除外されます。これには、direction が "in" または "both" に設定されたフィールドを含みます。

複数のフィルタを指定することもでき、これらは「統合」または「or」の演算子として動作します。

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

実数のストレージを持つ離散型のフィールドまたは整数のストレージを持つ離散型のフィールドは除外されます。

最初空のフィールド セットにフィールドが含まれる場合、include 文の順序は通常、フィールドが含まれる順序に影響を与えません。フィールド セット プロバイダのフィールドは、各条件に対して普通の順序で評価され、フィールド セットに含むかどうかを指定します。

役割

役割は、データ モデル出力フィールドに保持されるデータの種類を説明します。役割は AddField 要素で指定し、Condition 要素でテストすることができます。

役割は次の通りです。

テーブル 4-12
モデル出力の役割

役割	意味
unknown	役割を指定されていません（指定することはできません）。
predictedValue	このフィールドには、対象フィールドの予測値が含まれます。
probability	予測の確率または確信度。
residual	残差の値。
standardError	予測の標準偏差。
entityId	エンティティ ID。通常、クラスタ モデルのクラスタ ID を示します。
entityAffinity	エンティティの類似性。通常、モデルのクラスタの中心からの距離を示します。
upperConfidenceLimit	予測の確信度の上限。
lowerConfidenceLimit	予測の確信度の下限。
propensity	傾向スコア。追加の tag 属性は行傾向を参照するか調整済み傾向を参照するかを指定します。
value	別の出力に関連するモデルの値を示します（下記参照）。
supplementary	他の役割でカバーされていないモデルで生成された情報。

value の例として、異常値検出モデルで、1 つのフィールドがフィールド名を表し、もう 1 つがフィールドの異常度を示す尺度を指定する、各グループが 2 つのフィールドで構成されるフィールドのグループを生成します。この場合、value はフィールド名となります。

論理演算子

多くの要素では、論理演算子 And、Or および Not を使用して、たとえば複合条件を設定する場合など、さまざまな処理を指定することができます（「複合条件」（ p. 103 ）を参照してください）。

書式

And 要素の形式は次のとおりです。Or 要素および Not 要素の形式はほぼ同じですが、囲むタグがそれぞれ <Or>...</Or> と <Not>...</Not> となる点が異なります。

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

Condition 要素は、テストされる条件を指定します。 詳細は、p. 98 Conditions を参照してください。

And、Or および Not 子要素は入れ子にすることができます。

Conditions

いくつかのオブジェクトの動作を条件を使用して変更することができます。Condition 要素（IF 文と同等）で指定します。たとえば、コマンドで実行されるノードは、プロパティに特定の値がある場合特定のオプションのみを含むなど、実行情報に条件を追加することができます。同様に、ユーザーインターフェイスのプロパティ コントロールを、別のコントロールに特定の値が含まれている場合にのみ有効化または表示することができます。

条件は単純にすることも、複合にすることもできます。**単純な条件**は、次の内容で構成されています。

- 値のソース（プロパティまたはコントロールのいずれか）
- テスト
- オプションのテスト値

複合条件を使用すると、他の条件と結合して複雑な論理条件を形成することができます。複合条件で使用されるのは次のとおりです。

- And

- Or
- Not

書式

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

ここでの意味は次の通りです。

container は、値が条件でテストされる特定のコンテナの名前を指定します。

control は、条件で値がテストされるプロパティ コントロールを指定します。prop_name は、コントロールが定義される要素の property 属性の値です（ダイアログ タブのプロパティ パネルなど）。

property は、条件で値がテストされるプロパティを指定します。name は、コントロールが定義されるProperty 要素の name 属性の値です。

op は、条件の演算子です。 詳細は、 p. 99 条件演算子 を参照してください。

value は、条件でテストされる特定の値です。

例

条件設定の例は、単純な条件 p. 102 および 複合条件 p. 103 を参照してください。

条件演算子

一連の演算子は、ほとんどの条件を処理するために使用できます。

テーブル 4-13
値でサポートされるテスト

演算子	値	説明
equals	value	プロパティが提供された値と等しい場合は真です（文字列値の場合は大文字/小文字を区別します）。
notEquals	value	プロパティが提供された値と等くない場合は真です（文字列値の場合は大文字/小文字を区別します）。
in	値のリスト	プロパティは提供された値のリスト内になる場合は真です。

テーブル 4–14
数値でサポートされるテスト

演算子	値	説明
lessThan	number	プロパティは提供された数値を下回る場合は真です。
lessOrEquals	number	プロパティは提供された数値以下の場合は真です。
greaterThan	number	プロパティは提供された数値を上回る場合は真です。
greaterOrEquals	number	プロパティは提供された数値以上の場合は真です。

テーブル 4–15
文字列値でサポートされるテスト

演算子	値	説明
isEmpty	–	プロパティに 0 の長さの文字列が含まれる場合は真です。
isNotEmpty	–	プロパティに 0 以外の長さの文字列が含まれる場合は真です。
startsWith	string	プロパティが提供された文字列で開始する場合は真です（大文字/小文字を区別）。
startsWithIgnoreCase	string	プロパティが提供された文字列で開始する場合は真です（大文字/小文字の区別はなし）。
endsWith	string	プロパティが提供された文字列で終了する場合は真です（大文字/小文字を区別）。
endsWithIgnoreCase	string	プロパティが提供された文字列で終了する場合は真です（大文字/小文字の区別はなし）。
equalsIgnoreCase	string	プロパティが提供された文字列と等しい場合は真です（大文字/小文字の区別はなし）。
hasSubstring	string	プロパティが提供された文字列を含む場合は真です（大文字/小文字を区別）。
hasSubstringIgnoreCase	string	プロパティが提供された文字列を含む場合は真です（大文字/小文字の区別はなし）。
isSubstring	string	プロパティが提供された文字列のサブ文字列である場合は真です（大文字/小文字を区別）。
isSubstringIgnoreCase	string	プロパティが提供された文字列のサブ文字列である場合は真です（大文字/小文字の区別はなし）。

テーブル 4-16
リスト プロパティでサポートされたテスト

演算子	値	説明
isEmpty	–	リスト内の項目数が 0 の場合は真です。
isNotEmpty	–	リスト内の項目数が 0 でない場合は真です。
countEquals	number	リスト内の項目数が提供された値と等しい場合は真です。
countLessThan	number	リスト内の項目数が提供された値を下回る場合は真です。
countLessOrEquals	number	リスト内の項目数が提供された値以下の場合は真です。
countGreaterThan	number	リスト内の項目数が提供された値を上回る場合は真です。
countGreaterOrEquals	number	リスト内の項目数が提供された値以上の場合は真です。
contains	value	提供された項目がリスト内にある場合は真です。

テーブル 4-17
フィールド プロパティでサポートされたテスト

演算子	説明
storageEquals	ストレージが提供された値と等しい場合は真です。
typeEquals	種類が提供された値と等しい場合は真です。
directionEquals	フィールドの役割（direction）が提供された値と等しい場合は真です。
isMeasureDiscrete	フィールドのデータ型が discrete の場合は真です（set、flag または orderedSet のいずれかのみ）。
isMeasureContinuous	フィールドのデータ型が range の場合は真です。
isMeasureTypeless	フィールドのデータ型が typeless の場合は真です。
isMeasureUnknown	フィールドのデータ型が unknown の場合は真です。
isStorageString	フィールドのストレージ タイプが string の場合は真です。
isStorageNumeric	フィールドのストレージ タイプが numeric の場合は真です。
isStorageDatetime	フィールドのストレージ タイプが datetime の場合は真です。
isStorageUnknown	フィールドのストレージ タイプが unknown の場合は真です。

演算子	説明
isModelOutput	フィールドがモデル出力フィールドの場合は真です。
modelOutputRoleEquals	フィールドの役割が次の表で示される有効な役割のいずれかである場合は真です。
modelOutputTargetFieldEquals	対象フィールドが指定された値（文字列）と等しい場合は真です。
modelOutputHasValue	フィールドはモデル出力フィールドで、それに関連する値を持つ場合は真です。
modelOutputTagEquals	タグが指定された値（文字列）と等しい場合は真です。

キー プロパティ をサポートする条件演算子は次のとおりです。

- isEmpty
- isNotEmpty
- countEquals
- countLessThan
- countLessOrEquals
- countGreaterThan
- countGreaterOrEquals
- contains

単純な条件

単純な条件は、テストされる最初の値のソース（プロパティ名またはコントローラ名のいずれか、または評価される式）、実行されるテスト、オプションでテストが実行される値で構成されます。

例

次の例では、values_grouped というブール プロパティが真である場合、真と評価します。

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

次の例では、ブール値を表示する values_grouped というコントロールがチェックされた場合に真と評価されます。

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

次の例は、plot_fields というリスト プロパティに少なくとも 1 つの値がある場合に真と評価します。

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

次の例では、input_fields というリスト プロパティにインスタンスかされた値だけが含まれている場合に真と評価します。

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

最後の例では、input_fields というプロパティにインスタンス化されていないフィールドを示す値を少なくとも 1 つ含む場合に真と評価します。

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

複合条件

単純な条件のグループを、論理演算子を使用して結合することができます。

例

次の例では、ブール values_grouped プロパティが真で、group_fields に少なくとも 1 つの値が含まれる場合に真と評価します。

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

次の例では、ブール values_grouped プロパティが真、または group_fields に少なくとも 1 つの値が含まれる場合に真と評価します。

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

次の例では、group_fields に少なくとも 1 つの値が含まれている場合に真と評価します。

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

複合条件をネストして、条件の組み合わせを提供することができます。

スクリプトの CLEF ノードの使用

Node 要素の scriptName 属性を使用してスクリプトの CLEF ノードを参照できます。同じように、Property 要素の scriptName 属性を使用して、スクリプトのノードのプロパティを参照できます。

いずれの場合も、scriptName 属性はオプションですが、拡張またはプロパティ間の名前の競合を回避するために属性を使用することをお勧めします。

ノード定義でスクリプト名を省略すると、スクリプトは拡張名が最初に付いている id 属性の値によってノードを参照できます。たとえば、 ID import によってデータ リーダー ノードを定義する myext という名前の拡張子が指定されている場合、スクリプトはノードを myextimport として参照することができます。

プロパティ 定義でスクリプト名を省略すると、スクリプトは name 属性の値を使用してプロパティを参照できます。

詳細は、『IBM® SPSS® Modeler Scripting and Automation Guide』を参照してください。

例 – ノードの編集および実行

次の例では、「データ リーダー ノード (Apache Log Reader)」 (p. 36)で説明しているデータ リーダー ノードの例を編集および実行するタスクを自動化するスクリプトの使用方法について説明しています。

Apache Log Reader ノードの特性ファイルでは、ノードの設定は次のように始まります。

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

スクリプトでは、ノードとプロパティを次のように参照します。

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

ここで、installation_directory は、IBM® SPSS® Modeler がインストールされているディレクトリです。

このスクリプトを実行すると、次のようになります。

- データ リーダー ノードを作成する
- combined_log_format.txt を読み取る Apache ログ ファイルとして指定する
- テーブル ノードを作成する

- データ リーダー ノードをテーブル ノードを接続する
- テーブル ノードを実行する

例 – キー プロパティ

キー プロパティは、標準的なスクリプト シンタックスをサポートします。たとえば、「構造化プロパティ」（ p. 85 ）の最初の例のプロパティタイプで示す構造は、次のように定義できました。

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

1 つの属性を次のように変更できました。

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

下位互換性の保持

既存の拡張に更新を計画する場合、その拡張の以前配布されたバージョンとの互換性を保持してください。変更によっては不都合な影響のないもの、重大なリスクを伴うもの、また互換性を壊すため回避するべきものがあります。

リスクのない変更

次の変更は、下位互換性には影響ありません。

- Node、ModelOutput、DocumentOutput または InteractiveModelBuilder 要素の追加
- これらの要素への新しい Property 定義と関連する新しいコントロールの追加
- これらの要素への新しいコンテナの追加*
- 既存の列挙型プロパティへの新しい値の追加

* これらの新しいコンテナを使用するコードは、拡張の以前のバージョンで作成されたオブジェクトの新しいコンテナが空となるようにする必要があります。

重大なリスクのある変更

既存の宣言に対する変更は、互換性を壊す重大なリスクを引き起こします。配布する前に慎重にテストする必要があります。

回避するべき変更

次の変更は、互換性を壊すため、回避する必要があるとされています。

- ExtensionDetail 要素の id 属性または providerTag 属性の値の変更
- Node 要素、ModelOutput 要素、DocumentOutput または InteractiveModelBuilder 要素の id 属性の値の変更
- Node 要素、ModelOutput 要素、DocumentOutput 要素または InteractiveModelBuilder 要素の拡張からの削除
- Property または PropertyType 要素の valueType 属性の値の変更

章 5

モデルおよびドキュメントの構築

モデルおよびドキュメント構築の概要

標準 IBM® SPSS® Modelerモジュールにはユーザーが様々なモデルおよびグラフを生成（または「構築」）できるノードが含まれています。CLEF で標準的に提供されていないその他のモデルやドキュメント（グラフおよびレポート）を構築する追加ノードを定義することができます。

モデル ビルダおよびドキュメント ビルダ ノードを定義する場合、これらのノードを実行する際に作成されるオブジェクトも定義する必要があります。「コンストラクタ」と呼ばれる項目によって実行します。

次の項では、このプロセスについて詳細に説明します。

[モデル]

モデルは、一連の入力フィールドに基づいて結果を予測するために使用できるルールのセット、式、または方程式です。結果を予測する機能は、予測分析の中心となる目標です。IBM® SPSS® Modeler で結果を予測するために、次のことを行います。

- 既存データからモデルを生成
- 生成されたモデルをデータに適用し、予測を実行

モデル生成のプロセスは、モデルの「構築」とも呼ばれ、SPSS Modeler ではモデル作成ノードを使用してモデルを構築します。CLEF で、モデル作成ノードはノードの定義に使用される XML 分の構文から取得された名前である **モデル ビルダー ノード**として参照されます。 詳細は、 2 章 p.12 モデル ビルダー ノード を参照してください。

モデルをデータに適用するプロセスは、「データのスコアリング」とも呼ばれています。データのスコアリングを行うと、モデル作成から取得した情報を使用して、新規レコードの予測を作成できます。SPSS Modeler では、生成されたモデルのアイコンをストリーム領域に追加してデータのスコアリングを行います。アイコンは金色のナゲットの形をしているため、SPSS Modeler の生成されたモデルは「モデル ナゲット」として参照されます。CLEF では、マネージャ領域の［モデル］タブのモデル ナゲットは**モデル出力オブジェクト**と呼ばれ、領域に追加されると **モデル アプライヤ ノード** として認識されます。 詳細は、 2 章 p.15 モデル アプライヤ ノード を参照してください。

© Copyright IBM Corporation 1994, 2012.

文書

グラフまたはレポート出力など、モデル以外のオブジェクトの生成が必要な場合があります。IBM® SPSS® Modeler では、モデル以外のオブジェクトは**ドキュメント**とも呼ばれ、**ドキュメント ビルダー ノード**によって生成されます。 詳細は、 2 章 p.14 ドキュメント ビルダー ノード を参照してください。

Constructors

コンストラクタは、ストリーム内のノードの実行またはストリームへのオブジェクトの生成の結果として作成されるオブジェクトを定義します。

コンストラクタは次のいずれかに定義できます。

- モデル ビルダー ノード
- ドキュメント ビルダー ノード
- モデル アプライヤ ノード
- モデル出力オブジェクト

モデル ビルダー または **ドキュメント ビルダー** ノードの場合、コンストラクタを使用すると、これらのノードによりノード実行時の出力オブジェクトの生成方法を定義できます。出力オブジェクトの定義には複数のプロパティおよびコンポーネントが含まれる場合があり、Constructors セクションはこれらがどのように初期化されるか、または実行によって生成されたオブジェクトからどのように作成されるかを定義します。

モデル アプライヤ ノードの場合、コンストラクタはノードがストリームまたは［モデル］タブに生成できるオブジェクトの種類を定義します。

モデル出力オブジェクトいコンストラクタが定義された場合、以下のことを実行できます。

- モデル出力オブジェクトがストリーム領域にある場合に作成するモデル アプライヤ ノードを指定
- モデル出力オブジェクトの作成に使用された設定でモデル ビルダー ノードを生成

詳細は、 p.137 コンストラクタの使用 を参照してください。

モデルの構築

モデルを生成できるノード（モデル ビルダー ノード）を指定する場合、モデルを実際に作成するプロセスである、IBM® SPSS® Modeler のモデル ビルダー コンポーネントとのノードの通信方法を定義する必要があります。設定ファイルの Node 要素の定義で行います。

ノードが指定されていない場合、エンド ユーザーがモデル ビルダー ノードのダイアログ ボックスの［**実行**］ボタンをクリックするとすぐにモデル構築が開始されます。ただし、エンド ユーザーがモデルが実際に構築される前に［**実行**］をクリックした後でデータ値を調整または変更できるよう、**インタラクティブ モデル**定義することもできます。さらにインタラクティブ モデルの構築では、双方向性が定義される設定要素を使用する必要があります。 詳細は、 p. 120 インタラクティブ モデルの構築 を参照してください。

モデル ビルダー ノードを定義する場合、Node 要素に必要なものは次のとおりです。

- type="modelBuilder" 属性
- ModelBuilder 子要素
- CreateModelOutput 要素を含む Constructors 子要素（コンストラクタの使用 p. 137 を参照）

Node 要素の設定の形式ついては、ノード p. 66 を参照してください。

注: 後続のセクションの要素の定義（通常**形式**の見出しで識別）では、「(必須)」と指定されていない限り、要素の属性および子要素はオプションです。要素のシンタックスの詳細は、A 付録 の「CLEF XML スキーマ」(p. 280) を参照してください。

モデル構築において、拡張には生成されたモデルを説明するために ModelOutput 要素も必要です（モデル出力 p. 119 を参照）。ModelOutput 要素には、CreateModelApplier 定義を含む Constructors 子要素が必要です。 詳細は、 p. 140 モデル アプライヤの作成 を参照してください。

モデル ビルダー

ModelBuilder 要素は、モデル ビルダー ノードの動作を定義します。要素の属性および子要素によって定義します。

書式

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

```
   <ModelGeneration .../>
   <ModelFields .../>
   <AutoModeling .../>
</ModelBuilder>
```

ここでの意味は次の通りです。

- 入力フィールドがないまたは出力フィールドがないモデルをそれぞれ構築する場合、allowNoInputs および allowNoOutputs を明示的に使用する必要があります。
- false に設定されている場合、nullifyBlanks は、IBM® SPSS® Modeler のモデル ビルダー コンポーネントに渡されるデータの空白値をヌル値（$null$）に置き換える機能をオフにします。デフォルトでは、空白値をヌル値に置き換えますが、たとえばアルゴリズムがヌル値とは別に空白値を処理する必要がある場合に、この機能を無効化する必要があります。
- miningFunctions（必須）は、データ マイニング機能またはモデルを実行する機能を識別します。

テーブル 5-1
データ マイニング機能

機能	説明
classification	不明な対象値を持つレコードから、不明な対象属性の不連続な（set、flag、または orderedSet データ型）値を予測します。
approximation	不明な対象値を持つレコードから、不明な対象属性の連続した（range データ型）値を予測します。
clustering	類似したレコードのグループを識別し、それに応じてラベルを付けます。
association	データないの関連したイベントまたは属性を識別します。
sequence	時間構造データのシーケンス パターンを検索します。
分解	たとえば、下のデータ フィールドの内容を要約する派生フィールドを使用するなど、データの複雑さを軽減します。
conceptExtraction	テキスト マイニングに使用されます。
categorize	テキスト マイニングに使用されます。
timeSeries	過去のデータのパターンから将来の値を予測します。
anomalyDetection	クラスタ グループの平均からの偏差に基づいて、例外的なケースを検索します。
attributeImportance	対象属性に最も大きな影響を持つ属性を識別します。
supervisedMultiTarget	さまざまな確率の 1 つに対する（「はい」または「いいえ」）結果の尤度を推定します。

モデルが複数の機能を実行する場合、機能名は次の例のように大カッコ内でスペースで区切られます。

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

子要素

ModelBuilder 要素の子要素は次のとおりです。

テーブル 5-2
モデル ビルダーの宣言の子要素

子要素	説明	参照...
アルゴリズム	(必須) モデルの生成に使用されるアルゴリズムを指定します。	アルゴリズム p. 111
ModelingFields	User Interface セクションで一貫して使用される識別子を指定して、モデルの入力フィールドおよび出力フィールドのコントロールの場所を定義します。コントロール自体は、ModelingFields の InputFields 子要素および OutputFields 子要素に定義されます。	モデル フィールド p. 112
ModelGeneration	User Interface セクションで一貫して使用される識別子を指定して、生成されたモデルのモデル名コントロールの場所を定義します。	モデル生成 p. 116
ModelFields	モデルを持つデータのスコアリングに使用される入力フィールドおよび出力フィールドのセットを指定します。	モデル フィールド p. 116
AutoModeling	自動分類、自動クラスタまたは自動数値など、アンサンブル モデル作成ノードによってモデルを使用できるようにします。	自動化されたモデル作成 p. 125

アルゴリズム

Algorithm 要素は、モデルの生成に使用されるアルゴリズムの詳細を定義します。

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

ここでの意味は次の通りです。

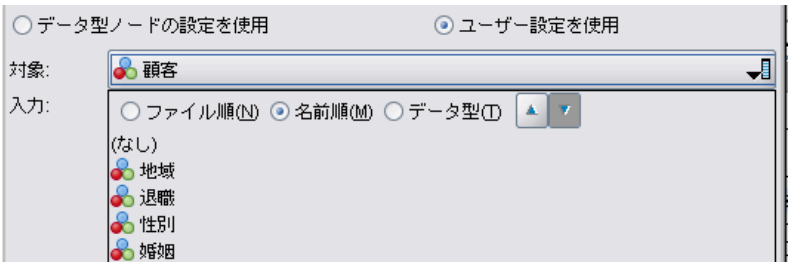
- value (必須) は、アルゴリズムの内部名です。設定ファイルのさまざまな場所で参照されます。 詳細は、 p. 117 モデル ビルダーの例 を参照してください。

- label（必須）は、アルゴリズムの説明です。
- labelKey は、ローカライズ用のラベルを識別します。

モデル フィールド

データ型ノードを使用してノードモデルの入力フィールドおよび出力フィールドを指定するのが、標準的な方法です。必要に応じて、フィールドの役割を [**入力**] または [**出力**] に設定します。モデル ビルダー ノードの場合、オプションで、ユーザーは上流のデータ型ノードの設定を上書きおよびカスタム設定を使用することができます。次に例を示します。

図 5-1
モデル入力および出力の指定



ModelingFields 要素によって行われます。この要素は、モデル ビルダー ノードの宣言の User Interface セクションで一貫して使用される識別子を指定し、モデルの入力フィールドおよび出力フィールドのコントロールを定義します。コントロール自体は、InputFields および OutputFields 子要素によって定義されます。

書式

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

ここでの意味は次の通りです。

- controlsId（必須）は、モデル ビルダー ノードの宣言の User Interface セクションの SystemControls 要素で一貫して使用される識別子です。モデルの入力フィールドおよび出力フィールドのコントロールを含むノードのダイアログ ボックスのタブを識別します。
- true に設定されている場合（デフォルト）、ignoreBOTH は、**両方**に方向が設定されたフィールドがモデルに無視されるよう指定します。

InputFields 要素および OutputFields 要素は、入力フィールド p. 113 の最初の項で説明されています。

例

この例では、モデル ビルダー ノードのダイアログ ボックスの［フィールド］タブのモデル フィールド コントロールの使用について説明します。まず、識別子はコントロールのセットに指定されます。

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

modelingFields 識別子は、［フィールド］タブが定義された時点で、ノードのダイアログ ボックスの User Interface セクションで一貫して参照されます。

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

入力フィールド

InputFields 要素は、ユーザーがモデルの入力フィールド（予測フィールド）を選択できるフィールドのセットを定義します。

セットは、このノードで表示されるすべてのフィールドで構成されています。フィールドがこのノードの上流で詳細にフィルタリングされている場合、フィルタを通過したフィールドのみが表示されます。特定のストレージ タイプおよびデータ型のフィールドのみが選択できるよう指定することによって、リストをより詳細に制限することができます。

```
<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="label" labelKey="label_key"/>
```

2 つの属性を指定して、入力フィールドとして使用されるフィールドのリストを制限できます。次のリストから選択する必要があります。

- storage は、リスト内に表示されるフィールドのストレージ タイプを指定するリストのプロパティです。たとえば、storage="[integer real]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。可能なストレージ タイプのセットについては、データおよびストレージ タイプ p. 251 のテーブルを参照してください。
- onlyNumeric が true に設定されている場合、数値型ストレージのフィールドのみが一覧表示されます。
- onlySymbolic が true に設定されている場合、シンボル値のストレージ タイプのフィールド（文字列）のみが一覧表示されます。
- onlyDatetime が true に設定されている場合、日付と時間のストレージ タイプのみが一覧表示されます。

2 番目の属性を以下から選択する必要があります。

- types は、リスト内に表示されるフィールドのデータ型を指定するリストのプロパティです。たとえば、types="[range flag]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。使用できるデータ 型のセットは次のとおりです。

  range

  flag

  set

  orderedSet

  numeric

  discrete

  typeless
- onlyRanges が true に設定されている場合、範囲のデータ型のフィールドのみが一覧表示されます。
- onlyDiscrete が true に設定されている場合、不連続な（フラグ、セット、データ型不明）のデータ型のみが一覧表示されます。

たとえば、storage="[integer]" および types="[flag]" を指定するコントロールでは、フラグ型である整数フィールドのみがリスト内に表示されます。

残りの属性は次の通りです。

- property は、フィールド値を保存するために使用されるプロパティの識別子です。
- multiple は、フィールド値が列挙リストである（true）か、そうでないか（false）を指定します。

- label は、コントロールの表示名です。
- labelKey は、ローカライズ用のラベルを識別します。

出力フィールド

OutputFields 要素は、ユーザーがモデルの出力フィールド（対象フィールド）を選択できるフィールドのセットを定義します。

セットは、このノードで表示されるすべてのフィールドで構成されています。フィールドがこのノードの上流で詳細にフィルタリングされている場合、フィルタを通過したフィールドのみが表示されます。特定のストレージ タイプおよびデータ型のフィールドのみが選択できるよう指定することによって、リストをより詳細に制限することができます。

```
<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="label"
   labelKey="label_key"/>
```

2 つの属性を指定して、出力フィールドとして使用されるフィールドのリストを制限できます。次のリストから選択する必要があります。

- storage は、リスト内に表示されるフィールドのストレージ タイプを指定するリストのプロパティです。たとえば、storage="[integer real]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。可能なストレージ タイプのセットについては、データおよびストレージ タイプ p. 251 のテーブルを参照してください。
- onlyNumeric が true に設定されている場合、数値型ストレージのフィールドのみが一覧表示されます。
- onlySymbolic が true に設定されている場合、シンボル値のストレージ タイプのフィールド（文字列）のみが一覧表示されます。
- onlyDatetime が true に設定されている場合、日付と時間のストレージ タイプのみが一覧表示されます。

2 番目の属性を以下から選択する必要があります。

- types は、リスト内に表示されるフィールドのデータ型を指定するリストのプロパティです。たとえば、types="[range flag]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。使用できるデータ 型のセットは次のとおりです。

  range

  flag

  set

  orderedSet

numeric

discrete

typeless

- onlyRanges が true に設定されている場合、範囲のデータ型のフィールドのみが一覧表示されます。
- onlyDiscrete が true に設定されている場合、不連続な（フラグ、セット、データ型不明）のデータ型のみが一覧表示されます。

たとえば、storage="[integer]" および types="[flag]" を指定するコントロールでは、フラグ型である整数フィールドのみがリスト内に表示されます。

残りの属性は次の通りです。

- property は、フィールド値を保存するために使用されるプロパティの識別子です。
- multiple は、フィールド値が列挙リストである（true）か、そうでないか（false）を指定します。
- label は、コントロールの表示名です。
- labelKey は、ローカライズ用のラベルを識別します。

モデル生成

ModelGeneration 要素は、ファイルの別の場所で使用される識別子を指定し、生成されたモデルのモデル名コントロールを含むモデル ビルダー ノードのダイアログ ボックスのタブを定義します。

図 5-2
モデル名コントロール

形式は次のとおりです。

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

controlsId 属性は、モデル ビルダー ノードの設定の User Interface セクションの SystemControls 要素で一貫して使用される識別子です。設定に SystemControls 要素があるタブには、モデル名コントロールが含まれます。

モデル フィールド

ModelFields 要素を使用して、モデルによってデータをスコアリングするために使用する入力フィールドおよび出力フィールドのセットである**モデル署名**を構築します。

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

この場合、inputDirections および outputDirections は、モデル署名の構築方法を指定します。値はin、out または both にできます。

フィールド自体は、AddField 要素で指定されます。prefix 属性は、フィールド名に追加される接頭辞を指定し、モデルで生成されたフィールドを示します。たとえば、フィールド名が field1 で接頭辞が $S の場合、生成されたフィールドの名前は $S-field1 となります。 詳細は、 4 章 p. 89 フィールドの追加 を参照してください。

ForEach 要素を使用すると、反復が可能です。 詳細は、 4 章 p. 92 ForEach 要素による反復 を参照してください。

フィールド グループを使用して、モデルから複数の出力フィールドをグループ化して反復することができます。たとえば、$S-field1-1、$S-field1-2 など、出力フィールドに反復を示す接尾辞を使用することができます。フィールド グループを 1 津使用すると、同じセットのフィールドがモデル出力で複数回表示されます。 詳細は、 p. 118 フィールド グループの例 を参照してください。

Automodeling

AutoModeling 要素を使用すると、自動分類、自動クラスタまたは自動数値など、アンサンブル モデル作成ノードによってモデルを使用できるようにします。 詳細は、 p. 125 自動化されたモデル作成 を参照してください。

モデル ビルダーの例

次の例では、反復ノードの例の設定ファイルの完全な Model Builder セクションを示します（ モデル ビルダー ノード (Interaction) p. 38 を参照）。

```
<Node id="interaction.builder" type="modelBuilder" palette="modeling" label="相互作用">
   <ModelBuilder miningFunctions="[classification]">
      <Algorithm value="robd" label="Robert のアルゴリズム" />
      <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>
```

フィールド グループの例

この例は SLRM ノードから取得し、 2 つの新規フィールドのグループをモデル署名に追加して、モデルのスコアリング時に生成されたデータを含みます。各入力レコードについて、新規フィールドのデータは、max_predictions プロパティの値でユーザーが指定した回数スコアリングされます。

新しい 2 つのフィールドは次のとおりです。

- $S-target ： 対象フィールドの予測値が含まれます。
- $SC-target ： この予測の確率値が含まれます。

これら 2 つのフィールドをグループ化するために、ModelFields セクションで宣言された同じグループの識別子が割り当てられます。グループの識別子が AddField 要素の group 属性によって割り当てられます。

モデル ビルダー ノードの宣言には、以下が含まれています。

```
<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>
```

モデル アプライヤ ノードの宣言には、以下が含まれています。

```
<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>
```

対象フィールド名は campaign となり、ユーザーは max_predictions プロパティに対応するフィールドに 2 を入力します。モデル ビルダー ノードを実行すると、次のフィールドがモデルに追加されます。

- $S-campaign-1
- $SC-campaign-1
- $S-campaign-2
- $SC-campaign-2

モデル出力

ModelOutput 要素は、ストリーム実行後にマネージャ領域の［モデル］タブに表示されるモデル出力オブジェクトを説明します。

書式

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

ここでの意味は次の通りです。

- id（必須）は、生成されたモデルの一意の識別子です。
- label（必須）は、［モデル］タブに表示される、生成されたモデルの表示名です。
- labelKey は、ローカライズ用のラベルを識別します。

ModelOutput 要素内に含まれる場合がある子要素は次のとおりです。

テーブル 5-3
モデル出力の宣言の子要素

子要素	定義	参照...
ModelProvider	モデル出力を保持するコンテナ、および出力が PMML 形式かどうかを定義します。	モデル プロバイダ p. 71
Properties	生成されたモデルに使用されるプロパティ。	Properties p. 71
コンテナ	生成されたモデルの出力が格納されるコンテナ。	コンテナ p. 73
UserInterface	モデル出力オブジェクトの［モデル］タブおよび［設定］タブなど、生成されたモデルの出力を表示できるユーザー インターフェイス。	ユーザー インターフェイス p. 75
Constructors	生成されたモデルによって作成されたオブジェクト。	コンストラクタの使用 p. 137

例

```
<ModelOutput id="interaction.model" label="相互作用モデル">
   <Properties>
   </Properties>
   <Containers>
      <Container name="model" />
   </Containers>
   <UserInterface>
      <Tabs>
         <Tab label="Model">
            <TextBrowserPanel container="model" textFormat="plainText" />
         </Tab>
      </Tabs>
   </UserInterface>
   <Constructors>
      <CreateModelApplier type="interaction.applier">
         <SetContainer target="model" source="model" />
      </CreateModelApplier>
   </Constructors>
</ModelOutput>
```

インタラクティブ モデルの構築

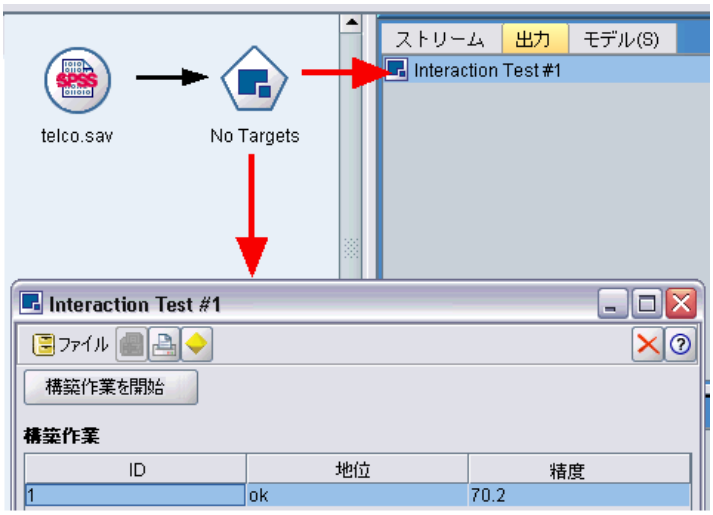
インタラクティブ モデルの構築は、モデル生成前にエンド ユーザーが対話処理できる出力オブジェクトを作成できる機能です。このインタラクティブ 出力オブジェクトは、マネージャ領域の［出力］タブに配置され、中間データセットが含まれます。中間データセットを使用して、モデルが生成される前にモデルを生成または単純化できます。インタラク

ティブ モデルの構築は、通常のモデル ビルダー ノードの設定に特別な要素を追加して実現されます。

- Node 定義の Constructors セクションには、CreateInteractiveModelBuilder 要素があります。
- 拡張には、専用の InteractiveModelBuilder 要素が含まれます。

対話ウィンドウというウィンドウを使用して、ユーザーは中間データセットと対話処理します。対話ウィンドウは出力オブジェクトが作成されるとすぐに表示されます。

図 5-3
インタラクティブ出力オブジェクトおよび対話ウィンドウ



対話処理は、拡張によって使用され、実行されるアルゴリズムに固有のものです。対話ウィンドウは、InteractiveModelBuilder 要素の User Interface セクションに定義されます。次のいずれかを指定して、対話ウィンドウを定義できます。

- ウィンドウを完全に定義するフレーム クラス（「User Interface セクション」（ p. 145 ）を参照）
- ウィンドウの各タブの、拡張オブジェクトパネルの属性として指定されるパネル クラス（「拡張オブジェクト パネル」（ p. 162 ）を参照）

ウィンドウを閉じた後、[出力] タブのオブジェクト名をダブルクリックして対話ウィンドウを再表示することができます。

対話ウィンドウの設定には、ユーザーが対話を終了した後モデルを生成するコードを使用する必要があります。説明されている例では、モデル生成の動作と関連する金色のナゲット アイコンを含むツールバー ボタンを使用して実行されています。このコードは 「インタラクティブ モデル構築の例」（ p. 124 ）の InteractiveModelBuilder セクションに表示されています。

インタラクティブ モデル ビルダーの作成

CreateInteractiveModelBuilder 要素は、ユーザーが対話処理する出力オブジェクトを説明します。CreateModelOutput 要素のインタラクティブ バージョンです。

書式

次のように、モデル ビルダー ノードの定義の Execution セクションで使用されます。

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

要素自体の形式は次のとおりです。

```
<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>
```

ここで、type（必須）は InteractiveModelBuilder 要素で作成された出力オブジェクトの識別子です。

Condition セクションを使用すると、条件を指定できます。 詳細は、4 章 p.98 Conditions を参照してください。

And、Or および Not の演算子を含む複雑な条件を指定することもできます。 詳細は、 4 章 p.98 論理演算子 を参照してください。

CreateModel 要素および CreateDocument 要素で、

- typeは、定義対象のモデルまたはドキュメントの識別子です。

- target（必須）は、モデルのコンテナの識別子です。このコンテナは、モデル出力セクションで定義されます。詳細は、 p. 119 モデル出力 を参照してください。
- sourceFile（必須）は、ノード実行中に生成される出力ファイルの識別子です。このファイルは、出力ファイル セクションで定義されます。 詳細は、 4 章 p. 78 出力ファイル を参照してください。

例

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

この例では、識別子 my.interaction を持つ出力オブジェクトが設定に InteractiveModelBuilder 要素が含まれるモデル ビルダー ノーづの実行時に作成されるよう指定しています。出力オブジェクト自体は、my.interaction 識別子を参照する InteractiveModelBuilder 要素によって設定ファイルの別の場所に定義されます。次に例を示します。

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

インタラクティブ モデル ビルダー

この要素は、インタラクティブ出力オブジェクトを定義し、エンド ユーザーは生成される前にモデルを調整または単純化できます。

InteractiveModelBuilder 要素は、 対応する CreateInteractiveModelBuilder 要素を含むモデル ビルダー ノードの定義に従います。 詳細は、 p. 122 インタラクティブ モデル ビルダーの作成 を参照してください。

書式

InteractiveModelBuilder 要素の形式は次のとおりです。

```
<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>
```

ここでの意味は次の通りです。

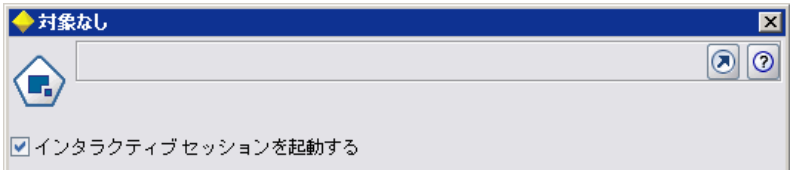
- id（必須）は、生成されたモデルの一意の識別子です。
- label（必須）は、[モデル] タブに表示される、生成されたモデルの表示名です。
- labelKey は、ローカライズ用のラベルを識別します。

Properties、Containers、UserInterface および Constructors 要素の詳細は、see Properties p. 71 ， コンテナ p. 73 ， User Interface セクション p. 145 、および コンストラクタの使用 p. 137 を参照してください。

インタラクティブ モデル構築の例

この例では、ユーザーがモデルの生成前に対話処理するかどうかを単純なチェック ボックスで選択できるよう、モデル ビルダー ノードをどのように定義するかについて説明しています。以下はその例です。

図 5-4
対話処理の選択



実際にこの動作を表示するには、このリリースで提供された Interaction のサンプルノードを使用してください。 詳細は、 3 章 p. 38 モデル ビルダー ノード（Interaction）を参照してください。

まず、モデル ビルダー ノードはブール プロパティを指定します。

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

ノード設定の User Interface セクションでは、[モデル] タブを定義するセクションにはこのプロパティへの参照が含まれます。

```
<Tab label="Model">
   <PropertiesPanel>
      <CheckBoxControl property="interactive" label="インタラクティブ セッションの開始" />
   </PropertiesPanel>
</Tab>
```

同じノードの CreateInteractiveModelBuilder セクションでは、プロパティの設定が検定され、true の場合はインタラクティブ出力オブジェクトが次のように作成されます。

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

参照される出力オブジェクトは、拡張のInteractiveModelBuilder セクションに定義されます。

```
<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>
```

モデル生成の操作は、ToolbarItem 要素に定義されたツールバー ボタンに制御されます。

ExtensionObjectPanel 要素の panelClass 属性を使用すると、対話ウィンドウのそれぞれのタブのユーザー インターフェイスを制御する Java クラスを指定します。

自動化されたモデル作成

IBM® SPSS® Modeler では、自動分類ノード、自動クラスタ ノード、または自動数値ノードなど、アンサンブル モデル構築ノードのグループを標準として提供します。これらのノードは、あらゆるモデルの同時構築を自動化

し、エンド ユーザーは結果を比較し、データの最良のモデルを選択できます。CLEF では、ModelBuilder 要素に指定されたモデルをこれらのアンサンブルモードで使用できる AutoModeling 要素を提供しています。

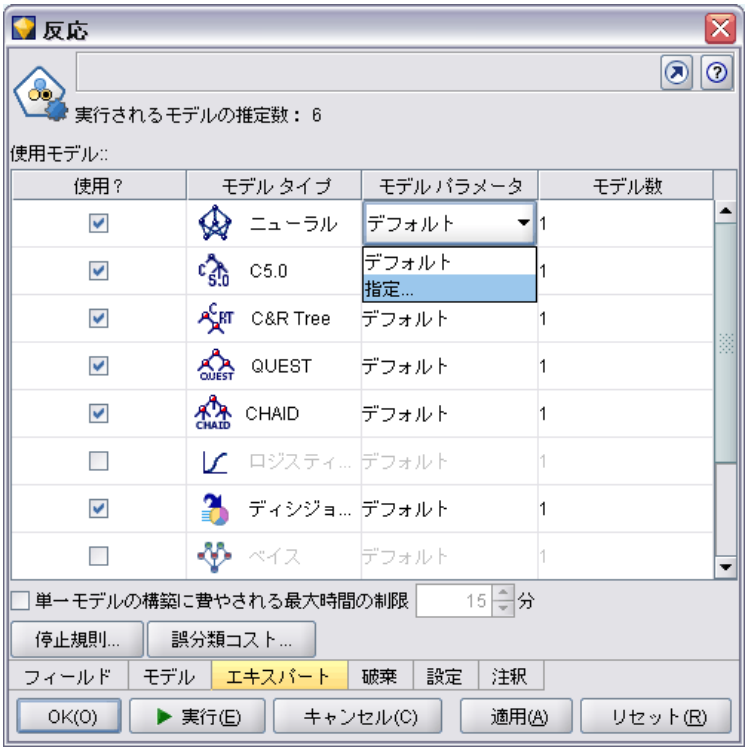
AutoModeling 要素の形式は次のとおりです。

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

この場合、enabledByDefault が、モデルをアンサンブル モデル作成ノードのデフォルトで使用できるかどうかを指定します（つまり、次の図では [使用?] 列が特定モデルの場合デフォルトでチェックされています)。この属性が省略されている場合、値 true が想定されます。

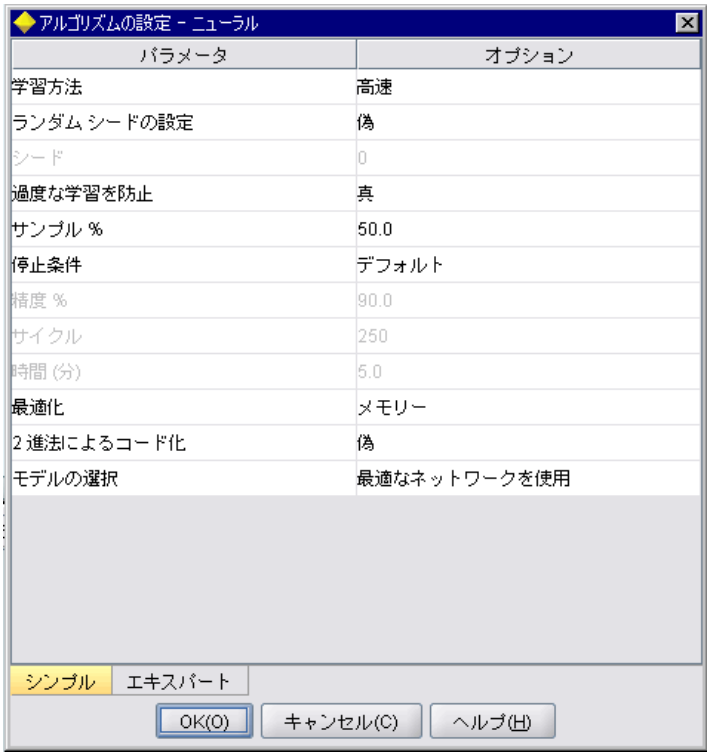
アンサンブル モデル構築ノードのダイアログ ボックスの［エクスパート］タブに、ユーザーが選択して構築できるモデルが表示されます。次に例を示します。

図 5-5
アンサンブル モデル構築ノードの [エキスパート] タブ



ユーザーがモデル タイプのオプションを選択できる場合、特定モデルの **[モデル パラメータ]** フィールドの **[指定]** をクリックすると、[アルゴリズム設定] ダイアログ ボックスが表示されます。

図 5-6
アンサンブル モデル構築ノードのアルゴリズム設定



[アルゴリズム設定] ダイアログ ボックスには[シンプル] タブ および [エキスパート] タブが含まれ、モデル構築ノードのシンプル実行モードおよびエキスパート実行モードに対応しています。[シンプル] タブおよび [エキスパート] タブの内容は、SimpleSettings 要素および ExpertSettings 要素に制御されます。これらの要素は後続のセクションで説明されています。

さらに、Constraint 要素を使用して条件を指定し、エンド ユーザーが [アルゴリズム設定] ダイアログ ボックスのパラメータを編集または何らかの方法で制限できるようにします。 詳細は、 p. 132 制約条件 を参照してください。

[アルゴリズム設定] ダイアログボックスのパラメータには、複数の値をとるものもあります。複数の値が指定される場合、アンサンブル ノードはパラメータ値の可能なすべての組み合わせに対するモデル構築を試みます。たとえば一般化線型モデルユーザーが 2 つの分布（正規分布およびガンマ分布）と 3 つのリンク関数（恒等式、対数、べき乗）を指定する場合、自動数値ノードはこれらのパラメータの可能な組み合わせに 1 つずつ、6 つの一般化線型モデルの構築を試みます。

シンプル設定

SimpleSettings 要素は、アンサンブル モデル構築ノードの一般化線型モデルの［アルゴリズム設定］ダイアログ ボックスの［シンプル］タブに表示されるパラメータを決定します。 詳細は、 p. 125 自動化されたモデル作成 を参照してください。

書式

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

PropertyGroup 要素（最低 1 つは必須）では次のようになります。

label は、プロパティ グループの表示ラベルで、グループの最初のパラメータの前にある小見出しとしてダイアログ ボックスに挿入されます。

labelKey は、ローカライズ用のラベルを識別します。label も labelKey も使用されていない場合、小見出しは挿入されません。

properties（必須）は、タブに表示されるプロパティのリストです。prop_name1、prop_name2 などの値は、このプロパティが定義される Property 要素の name 属性の値です。 詳細は、 4 章 p. 71 Properties を参照してください。

例

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

判別ノードのこの例では、関連するアンサンブル モデル構築ノード（この場合、自動分類ノード）内にある一般化線型モデルの［アルゴリズム設定］ダイアログ ボックスの［シンプル］タブに **方法**パラメータだけが表示されるようにします。label 属性または labelKey 属性が指定されていない場合、ダイアログ ボックスにパラメータの小見出しは表示されません。

エキスパート設定

ExpertSettings 要素は、アンサンブル モデル構築ノードの一般化線型モデルの［アルゴリズム設定］ダイアログ ボックスの［エキスパート］タブに表示されるパラメータを決定します。 詳細は、 p. 125 自動化されたモデル作成 を参照してください。

書式

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

true の場合、Condition 要素は、後続の PropertyGroup 要素または有効化される要素に識別されるパラメータを使用する条件を指定します。 詳細は、4 章 p.98 Conditions を参照してください。

PropertyGroup 要素（最低 1 つは必須）では次のようになります。

label は、プロパティ グループの表示ラベルで、グループの最初のパラメータの前にある小見出しとしてダイアログ ボックスに挿入されます。

labelKey は、ローカライズ用のラベルを識別します。label も labelKey も使用されていない場合、小見出しは挿入されません。

properties（必須）は、タブに表示されるプロパティのリストです。prop_name1、prop_name2 などの値は、このプロパティが定義される Property 要素の name 属性の値です。 詳細は、 4 章 p.71 Properties を参照してください。

例

次の例では、[アルゴリズム設定] ダイアログ ボックスの [エキスパート] タブに、最初は [**シンプル**] に設定される **モード** パラメータが含まれます。

図 5-7
エキスパート設定の無効化

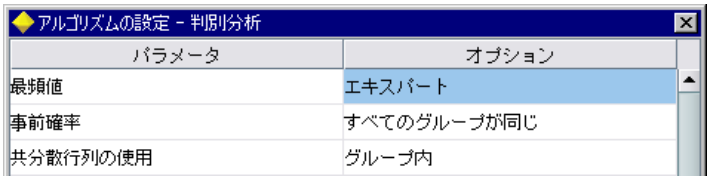
以下は、[**エキスパート**] に設定する **モード** パラメータが変更される場合にのみ、他の [エキスパート] タブのパラメータが有効化されるよう指定します。

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

```
</ExpertSettings>
```

[**エキスパート**] に設定する **モード** を変更すると、プロパティ グループの 2 つのパラメータが有効化されます。

図 5-8
エキスパート設定の有効化

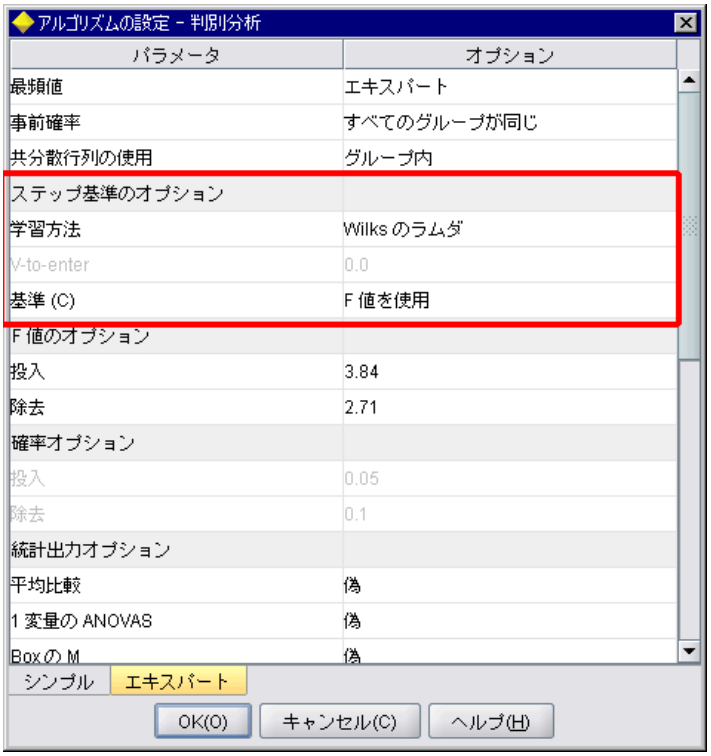


次の例では、プロパティ グループのラベルの使用を説明します。

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

この場合、PropertyGroup 要素は、説明中で強調表示されたパラメータを制御します。

図 5-9
エキスパート設定の無効化



labelKey 属性を使用すると、CLEF は拡張のプロパティ ファイル内の対応エントリから、プロパティ グループの小見出しに使用する表示テキストを取得できます。

automodel.stepping_criteria_options=Stepping Criteria option(s)

詳細は、 8 章 p. 231 プロパティ ファイル を参照してください。

制約条件

Constraint 要素は、アンサンブル モデル構築ノードのモデルの［アルゴリズム設定］ダイアログ ボックスに表示されたパラメータの編集、またはなんらかの制約を許可する制約を指定します。たとえば、エンド ユーザーが現在パラメータの変更を許可されていない場合、特定のパラメータを無効化できます。

書式

```
<Constraint property="prop_name" singleSelection="true_false">
   <Condition property="prop_name" op="operator" value="値"/>
```

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

ここでの意味は次の通りです。

- property （必須）は編集または制限するパラメータを識別します。prop_name はこのパラメータに対応するプロパティが定義されている Property 要素の name 属性の値です。 詳細は、 4 章 p.71 Properties を参照してください。
- singleSelection は、エンド ユーザーがパラメータに使用可能な値を複数選択できるかどうかを制御します。true に設定されていると、複数の値が［アルゴリズム設定］ダイアログ ボックスの該当するパラメータの［オプション］フィールドに表示されている場合であっても、選択できる値は 1 つだけです。false（デフォルト）に設定されていると、次の例で説明されているとおり、使用可能な値から 1 つまたは複数の値を選択できます。

Condition 要素は、実際の制約を指定します。 詳細は、 4 章 p.98 Conditions を参照してください。

And、Or、および Not 要素を使用して、複合条件を指定できます。 詳細は、 4 章 p.98 論理演算子 を参照してください。

例

この例は、一般化線型ノードの設定ファイルから取得しています。エキスパート モードの場合も、一部のパラメータはデフォルトでは有効化されていません。

図 5-10
制約の効果 – パラメータの無効化

この制約は、値パラメータが有効化される条件を指定します。

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

値パラメータが有効化される前に、**尺度パラメータ方法**パラメータ（scale_method プロパティで識別）は、[**固定値**] に設定し、**分布**は、[**正規分布**]、[**逆ガウス**] または [**ガンマ**] に設定する必要があります。

図 5-11
制約の効果 – パラメータの有効化

モデルの適用

モデルの適用とは、生成されたモデルを使用してデータをスコアリングする、つまり新規レコードの予測を作成するためのモデル構築から取得した情報を使用することです。IBM® SPSS® Modeler では、モデル アプライヤ ノードを使用してモデルを適用します。 詳細は、 2 章 p. 15 モデル アプライヤ ノード を参照してください。

設定ファイルのモデル アプライヤ ノードを定義すると、生成されたモデルを適用するためのフレームワークを作成します。SPSS Modeler では、モデル出力オブジェクトを示すアイコンをマネージャ領域の［モデル］タブからストリーム領域にドラッグして、モデル アプライヤ ノードのインスタンスを作成します。モデル アプライヤ ノードが定義されていない場合、モデル ビルダー ノードを実行すると未調整のモデルを生成します。このモデルをストリーム領域に追加することはできません。

モデル アプライヤ ノードを定義する場合、Node 要素に必要なものは次のとおりです。

- type="modelApplier" 属性
- CreateModelOutput 要素を含む Constructors 子要素（コンストラクタの使用 p. 137 を参照）

Node 要素の設定の形式ついては、ノード p. 66 を参照してください。

ドキュメントの構築

ドキュメント ビルダー ノードを定義する場合、Node 要素に必要なものは次のとおりです。

- type="documentBuilder" 属性
- DocumentBuilder 子要素

ドキュメント構築において、拡張には生成されたモデルを説明するために DocumentOutput 要素も必要です。 詳細は、 p. 119 モデル出力 を参照してください。

Node 要素の設定の形式ついては、ノード p. 66 を参照してください。

ドキュメント ビルダー

DocumentBuilder 要素は、ドキュメント ビルダー ノードの動作を定義します。定義では DocumentGeneration 子要素を使用し、ドキュメント生成コントロールを含むドキュメント ビルダー ノードのダイアログ ボックスのタブを指定する必要があります。コントロール自体は、User Interface セクションで定義されます（6 章の「ユーザー インターフェイスの構築」(p. 142) 参照）。

書式

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

この場合、controlsId（必須）は、ドキュメント生成コントロールが表示される場所を指定するシステム コントロールによって使用される識別子です。

例

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

ドキュメント出力

DocumentOutput 要素は、ストリーム実行後にマネージャ領域の［出力］タブに表示されるドキュメント出力オブジェクトを説明します。

書式

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

ここでの意味は次の通りです。

- id（必須）は、生成されたドキュメントの一意の識別子です。
- label（必須）は、［出力］タブに表示される、生成されたドキュメントの表示名です。
- labelKey は、ローカライズ用のラベルを識別します。

DocumentOutput 要素内に含まれる場合がある子要素は次のとおりです。

テーブル 5-4
ドキュメント出力の宣言の子要素

子要素	定義	参照...
Properties	生成されたドキュメントに使用されるプロパティ。	Properties p.71
コンテナ	生成されたドキュメントの出力が格納されるコンテナ。	コンテナ p.73
UserInterface	生成されたドキュメント出力が表示できるユーザー インターフェイス。	ユーザー インターフェイス p.75
Constructors	生成されたドキュメントによって作成されたオブジェクト。	コンストラクタの使用 p.137

例

```
<DocumentOutput id="webstatusreport">
   <Containers>
      <Container name="webstatusreportdata" />
   </Containers>
```

```
    <UserInterface>
      <Tabs>
        <Tab label="Advanced" labelKey="advancedTab.LABEL">
          <TextBrowserPanel container="webstatusreportdata" textFormat="html" />
        </Tab>
      </Tabs>
    </UserInterface>
</DocumentOutput>
```

コンストラクタの使用

Constructors 要素は、設定ファイル内の次の 2 つの場所のいずれかに使用できます。

- ノード定義の Execution セクション（モデルまたはドキュメント出力オブジェクトの場合）
- モデル出力定義（モデル アプライヤ ノードの場合）

1 つのノードで生成できる出力オブジェクトは 1 つだけです。この制限は、既存のノード既存のノードに対するスクリプトおよびクライアント API インターフェイスとの一貫性については必要ありません。。

書式

Execution セクションで使用される場合、Constructors 要素の形式は次のようになります。

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

モデル出力定義の場合、形式は次のようになります。

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

モデル出力の作成

この項では、[モデル] タブのモデル出力オブジェクトの作成方法、または [出力] タブのドキュメント出力オブジェクトの作成方法を定義します。

書式

```
<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>
```

CreateModelOutput 要素では、次のようになります。

- type（必須）は、Model Output セクションで定義されるモデル出力オブジェクトの識別子です。 詳細は、 4 章 p. 70 モデル出力 を参照してください。

Condition セクションを使用すると、条件を指定できます。 詳細は、 4 章 p. 98 Conditions を参照してください。

And、Or および Not の演算子を含む複雑な条件を指定することもできます。 詳細は、 4 章 p. 98 論理演算子 を参照してください。

CreateModel 要素および CreateDocument 要素で、

- typeは、定義対象のモデルまたはドキュメントの識別子です。
- target（必須）は、モデルのコンテナの識別子です。このコンテナは、モデル出力セクションで定義されます。詳細は、 p. 119 モデル出力 を参照してください。
- sourceFile（必須）は、ノード実行中に生成される出力ファイルの識別子です。このファイルは、出力ファイル セクションで定義されます。 詳細は、 4 章 p. 78 出力ファイル を参照してください。

例

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

ドキュメント出力の作成

この要素は、ドキュメント ビルダー ノードの定義の Execution セクションで使用され、作成されているドキュメント出力オブジェクトを識別します。

書式

```
<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>
```

type（必須）は、Document Output セクションで定義されるドキュメント出力オブジェクトの識別子です。 詳細は、 4 章 p. 70 ドキュメント出力 を参照してください。

Condition セクションを使用すると、条件を指定できます。 詳細は、 4 章 p. 98 Conditions を参照してください。

And、Or および Not の演算子を含む複雑な条件を指定することもできます。 詳細は、 4 章 p. 98 論理演算子 を参照してください。

CreateModel 要素および CreateDocument 要素で、

- typeは、定義対象のモデルまたはドキュメントの識別子です。
- target（必須）は、モデルのコンテナの識別子です。このコンテナは、モデル出力セクションで定義されます。詳細は、 p. 119 モデル出力 を参照してください。
- sourceFile（必須）は、ノード実行中に生成される出力ファイルの識別子です。このファイルは、出力ファイル セクションで定義されます。 詳細は、 4 章 p. 78 出力ファイル を参照してください。

例

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

インタラクティブ モデル ビルダーの作成

この要素は、インタラクティブ モデル ビルダー ノードの定義の Execution セクションで使用され、ユーザーが対話処理する出力オブジェクトを識別します。 詳細は、 p. 122 インタラクティブ モデル ビルダーの作成 を参照してください。

モデル アプライヤの作成

この要素は、モデル ビルダー ノードの定義の Model Output セクションの Constructors 要素内で使用されます（モデル出力 p. 119 を参照）。CreateModelApplier 要素は、モデル ビルダー ノードで生成されたモデル出力オブジェクトが領域内に配置された場合のモデル アプライヤ ノードの作成方法を定義します。

書式

```
<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>
```

CreateModelApplier 要素では、次のようになります。

- type（必須）は、作成されるモデル アプライヤ ノードの識別子です。実際このノードは、後でファイル内の Node ... type="modelApplier" 要素に定義されます。

Condition セクションを使用すると、条件を指定できます。 詳細は、 4 章 p. 98 Conditions を参照してください。

And、Or および Not の演算子を含む複雑な条件を指定することもできます。 詳細は、 4 章 p. 98 論理演算子 を参照してください。

CreateModel 要素および CreateDocument 要素で、

- typeは、定義対象のモデルまたはドキュメントの識別子です。
- target（必須）は、モデルのコンテナの識別子です。このコンテナは、モデル出力セクションで定義されます。詳細は、 p. 119 モデル出力 を参照してください。
- sourceFile（必須）は、ノード実行中に生成される出力ファイルの識別子です。このファイルは、出力ファイル セクションで定義されます。 詳細は、 4 章 p. 78 出力ファイル を参照してください。

例

次の例では、CreateModelApplier 要素に myapplier という名前の付いたモデル アプライヤ ノードへの前方参照が含まれます。このノードは、後続の Node 要素に定義されます。

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

章 6

ユーザー インターフェイスの構築

ユーザー インターフェイスについて

新しい CLEF ノードを追加する場合、エンド ユーザーが新しいノード、モデル出力、ドキュメント出力をどのように対話処理し、および拡張固有のノードをどのように適用するかを定義する必要があります。各オブジェクトのユーザー インターフェイスは、拡張の特性ファイルの UserInterface ファイルで定義されます。この章では、UserInterface セクションについて詳細に説明しています。

注： ファイルに定義されているオブジェクトの種類によって、1 つの特性ファイルに複数の UserInterface セクションがある場合があります。

CLEF オブジェクトのユーザー インターフェイスは、次のうち 1 つまたは複数で構成されています。

- メニュー エントリ
- パレット エントリ
- アイコン
- 1 つまたは複数のダイアログ ウィンドウ
- 1 つまたは複数の出力ウィンドウ

メニュー エントリおよび**パレット エントリ**を使用すると、IBM® SPSS® Modeler メニュー システムおよびノード パレットからそれぞれアクセスできます。**アイコン**は、メニュー、描画領域、およびさまざまなノード パレットのオブジェクトを識別します。**ダイアログ ウィンドウ**には、ストリームが実行される前にさまざまなオプションを指定できる、またオプションで実行か完了した場合に作成される出力を指定できるタブおよびコントロールが含まれます。**出力ウィンドウ**は、モデル出力（モデルをデータのセットに適用した結果など）またはドキュメント出力（グラフなど）を表示する場合に使用されます。

CLEF を使用すると、SPSS Modeler で提供された標準的なオブジェクトに新しい種類のオブジェクトを追加し、これらの新しいオブジェクトのユーザー インターフェイスを定義する一貫した手法をサポートできます。「ノード アイコンのデザイン」（ p.18 ）および 「ダイアログ ボックスのデザイン」（ p.23 ）では、CLEF でのアイコンおよびダイアログの作成に関するガイドラインが記載されています。

© Copyright IBM Corporation 1994, 2012.

アイコンは、特定のノードを識別するグラフィックの形式で、ノードの種類を識別する幾何学的図形の内部に表示されます。

ダイアログおよび**出力ウィンドウ**の特性は次のとおりです。

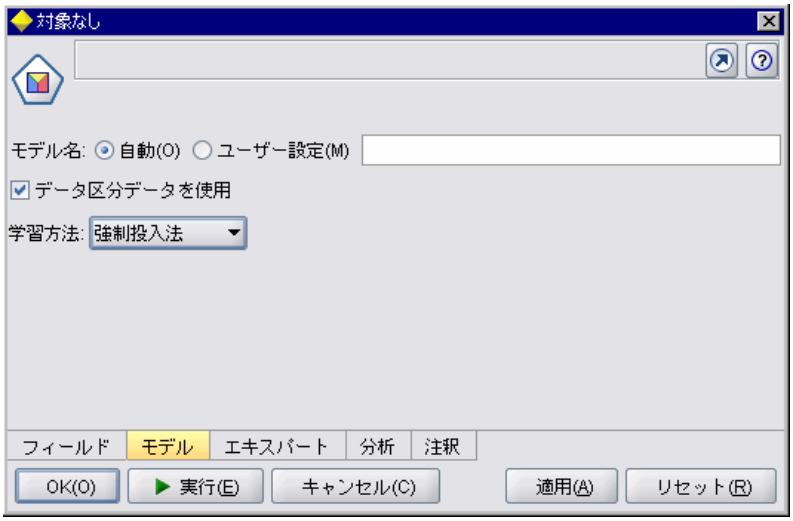
- 縮小アイコンおよびオブジェクトのタイトルが表示されたタイトル バー
- メニュー バーに表示されるのは次のとおりです。
  - メニュー
  - オブジェクト固有の操作ボタン
  - 一般的な操作ボタン（[最大化] または [ヘルプ]）
- メイン コンテンツ領域
- UI コンポーネントを論理グループに編成する複数のタブ
- サイズ変更機能

タブを使用して、ウィンドウのメイン コンテンツ領域をさまざまな方法で変更します。ダイアログ ウィンドウの場合、さまざまなタブにオブジェクト プロパティの異なるセットのコントロールが表示されます。コントロールは変更でき、ユーザーが [**適用**] または [OK] ボタンをクリックするとその結果が基本オブジェクトに適用されます。

出力ウィンドウの場合、タブを使用するとユーザーは結果の要約を表示または注釈を結果に追加するなど、出力に関連するさまざまな操作を実行することができます。

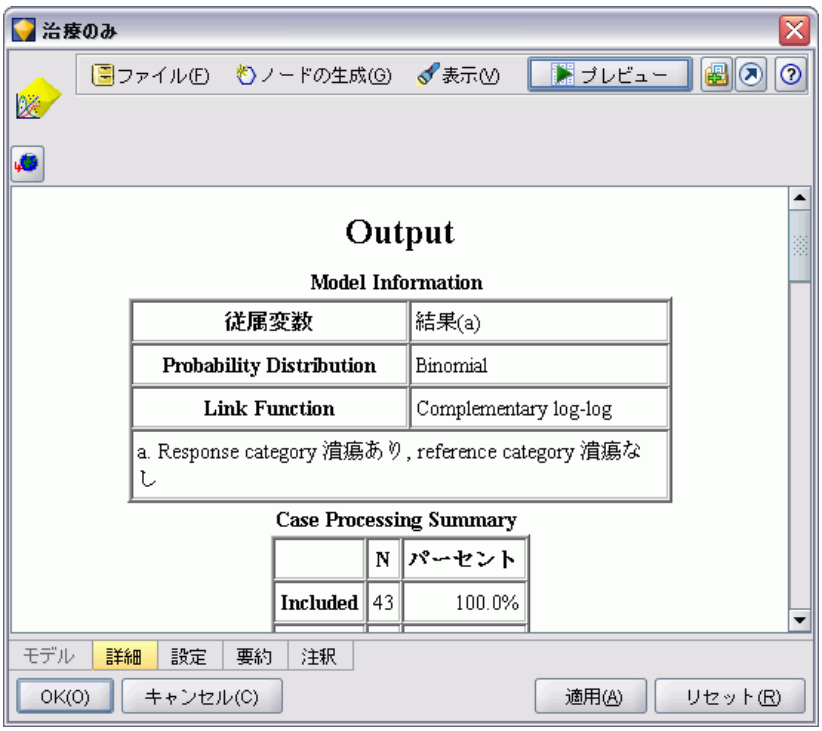
以下に、さまざまなタブおよびコントロールを含む単純なダイアログ ウィンドウの例を示します。

図 6-1
タブ化されたダイアログ



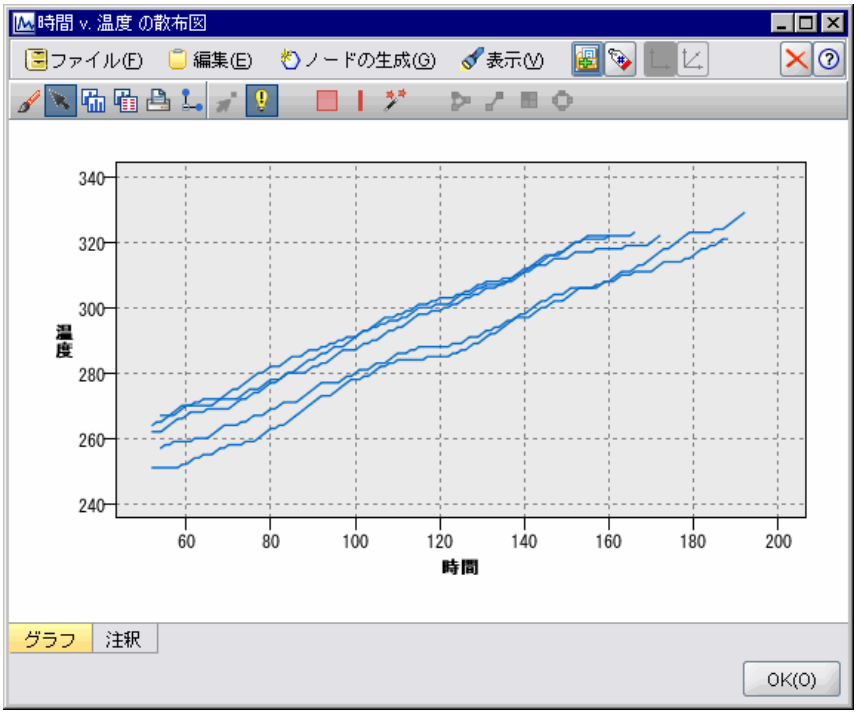
モデル出力ウィンドウの例を次に示します。

図 6-2
モデル出力ウィンドウ



最後に、ドキュメント出力ウィンドウ（この場合、グラフ）の例を示します。

図 6-3
ドキュメント出力ウィンドウ



User Interface セクション

オブジェクトのユーザー インターフェイスは特性ファイルのオブジェクト定義内の UserInterface 要素で宣言されています。ファイルに設定されているオブジェクト（モデル ビルダー ノード、モデル出力オブジェクト、モデル アプライヤ ノード）の数によって、同一特性ファイルには複数の UserInterface 要素が存在する場合があります。

それぞれの User Interface セクション内で定義できるのは次のとおりです。

- 領域またはパレットに表示するアイコン
- ダイアログまたは出力ウィンドウに表示するコントロール（カスタム メニューおよびツールバーの項目）
- プロパティ コントロールのセットを定義するタブ（ダイアログまたは出力ウィンドウ）

注 ： 後続の要素の定義（通常**形式**の見出しで識別）では、「(必須)」と指定されていない限り、要素の属性および子要素はオプションです。すべての要素のシンタックスの詳細は、A 付録 の「CLEF XML スキーマ」(p.280) を参照してください。

各ユーザー インターフェイスについて、実行されるプロセスを指定します。オプションである操作ハンドラまたはフレーム クラス属性のいずれかによって実行できます。操作ハンドラおよびフレーム クラス属性のいずれも指定されていない場合、実行されるプロセスはファイル内の別の場所で指定されます。

書式

UserInterface 要素の基本形式は次のとおりです。

```
<UserInterface>
  <Icons>
    <Icon .../>
     ...
  </Icons>
  <Controls>
    <Menu .../>
    <MenuItem .../>
     ...
    <ToolbarItem .../>
     ...
  </Controls>
  <Tabs>
    <Tab .../>
     ...
  </Tabs>
</UserInterface>
```

操作ハンドラは、カスタム操作を標準 IBM® SPSS® Modeler ウィンドウに追加する場合に使用されます。操作ハンドラは、ユーザーがノード ダイアログ、モデル出力ウィンドウ、またはドキュメント出力ウィンドウのカスタム メニュー オプションまたはツールバー ボタンを選択する場合に呼び出される Java クラスを指定します。ExtensionObjectFrame クラスまたは ActionHandler クラスのいずれかを実装します。いずれかの場合で、標準メニュー、タブ、およびツールバー ボタンなどの標準ウィンドウ コンポーネントが自動的に使用されます。 詳細は、 9 章 p.241 クライアント側 API クラス を参照してください。

操作ハンドラの形式は、次のような最初の行を除いて、基本形式と同じです。

```
<UserInterface actionHandler="Java_class" >
  ...
```

```
</UserInterface>
```

この場合、actionHandler は操作ハンドラの Java クラス名を表します。

フレーム クラスは、拡張によって標準 SPSS Modeler ウィンドウをカスタマイズするのではなく、独自のウィンドウを提供する場合、モデル出力オブジェクトまたはドキュメント出力オブジェクトに対して使用されます。フレーム クラスは、ウィンドウ全体およびプロセスを完全にしていする Java クラスです。自動的に使用される標準ウィンドウ コンポーネントはありません。クラスはこれらを個々に指定する必要があります。フレーム クラスは、ノードに対してではなくモデル出力オブジェクトまたはドキュメント出力オブジェクトに対してのみ使用でき、常に SPSS Modeler ダイアログを使用します。 詳細は、 p.223 カスタム出力ウィンドウ を参照してください。

フレーム クラスの形式は次のとおりです。

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

この場合 frameClass は、モデル出力オブジェクトまたはドキュメント出力オブジェクトのフレーム クラス名を表します。アイコン、コントロール、およびタブはフレームクラス自体で指定されるため、それらの要素はこの形式では使用されません。

UserInterface 要素の子要素は、後続のセクションで記述されます。

例

最初の例では、モデル ビルダー ノードのユーザー インターフェイスを示します。

```
<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>
```

モデル出力オブジェクトの対応するセクションは次のとおりです。

```
<UserInterface>
  <Icons>
    <Icon type="standardWindow" imagePath="images/browser_discriminant.gif" />
  </Icons>
  <Tabs>
```

```
      <Tab label="詳細" labelKey="advancedTab.LABEL"
       helpLink="discriminant_output_advancedtab.htm">
         <ExtensionObjectPanel id="DiscriminantPanel"
          panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
      </Tab>
   </Tabs>
</UserInterface>
```

モデル アプライヤー ノードの User Interface セクションは次のようになります。

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

アイコン

この項では、オブジェクトに関連するアイコンを定義します。

ノードの場合、この項では次の 2 つの Icon 要素を定義する必要があります。

- 領域に表示するための大きいバージョン
- 領域に表示するための小さいバージョン

モデル出力およびドキュメント出力の場合、ウィンドウ フレークの左上部に表示される縮小アイコンが定義されます。

ノード アイコンのデザイン p. 18 では、CLEF にアイコンを作成するためのガイドラインが記載されています。

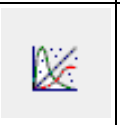
書式

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

ここでの意味は次の通りです。

type（必須）は、次のアイコンの種類のいずれかを表します。

テーブル 6-1
アイコンの種類

種類	例	説明
standardNode		領域内にノードの形状で表示される大きいサイズ（49 x 49 ピクセル）のアイコン。
smallNode		領域内にノードの形状で表示される小さいサイズ（38 x 38 ピクセル）のアイコン。
window		ブラウザまたは出力ウィンドウに表示される縮小（16 x 16 ピクセル）アイコン。

imagePath（必須）は、このアイコンで使用されるイメージの場所を識別します。特性ファイルがインストールされているディレクトリに関連した場所が指定されます。

例

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

コントロール

この項は、特性ファイルの Common Objects セクションで宣言される操作を起動するために使用するカスタム メニューおよびツールバーの項目を定義します。 詳細は、 4 章 p.55 Actions を参照してください。

書式

```
<Controls>
   <Menu .../>
    ...
   <MenuItem .../>
    ...
   <ToolbarItem .../>
    ...
</Controls>
```

後続のセクションで、Menu、MenuItem、および ToolbarItem 要素が記述されます。

例

次の例では、［ノードの生成］メニュー、設定が行われるダイアログのツールバーに項目を追加します。いずれの項目も、フィールド作成ノードを生成する generateDerive という名前の以前定義された操作を実装します。

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

メニュー

カスタム メニューを定義して、標準メニューのいずれかを追加できます。

書式

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

ここでの意味は次の通りです。

id（必須）は、追加されたメニューの一意の識別子です。

label（必須）ユーザー インターフェイスに表示されるメニューの表示名です（showLabel が true に設定されている場合）。

labelKey は、ローカライズ用のラベルを識別します。

systemMenu（必須）は、カスタム メニューが追加される標準メニューを識別します。menu_name の値は、次のいずれかです。

- file
- edit
- 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

* メイン IBM® SPSS® Modeler ウィンドウに追加した場合にのみ有効

showLabel は、項目の表示ラベルをユーザー インターフェイスに表示するかどうかを指定します。

showIcon は、項目に関連するアイコンをユーザー インターフェイスに表示するかどうかを指定します。

separatorBefore は、区切り（たとえば、メニュー項目の水平線またはツールバー ボタンのスペースなど）を、新しい項目の前にメニューに追加するかどうかを指定します。

separatorAfter は、新しい項目の後に区切りをメニューに追加するかどうかを指定します。

offset は、新しい項目の位置を定義する、負ではない整数です。たとえば、0 は新しい項目を最初の項目として追加します（デフォルトでは最後に追加します）。

mnemonic は、このコントロールを有効化するために Alt キーと組み合わせて使用するアルファベットの文字です（たとえば、値 S を指定すると、ユーザーは Alt + S キーを使用してこのコントロールを有効化できます）。

mnemonicKey は、ローカライズ用の mnemonic を識別します。mnemonic も mnemonicKey も使用されない場合、mnemonic はこのコントロールに使用できません。 詳細は、 p. 157 アクセス キーとキーボード ショートカット を参照してください。

メニュー項目

カスタム メニューの項目を定義して、標準メニューまたはカスタム メニューのいずれかを追加できます。

書式

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

ここでの意味は次の通りです。

action（必須）は、Common Objects セクションで定義された操作の識別子で、このメニュー項目で実行される操作を指定します。

systemMenu は menu_name で識別される標準メニューに表示される次のいずれかの項目を指定します。

- file
- edit
- 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

* メイン IBM® SPSS® Modeler ウィンドウに追加した場合にのみ有効

customMenu は Menu 要素の識別子でメニュー項目がカスタム メニューに表示されるように指定します。

showLabel は、項目の表示ラベルをユーザー インターフェイスに表示するかどうかを指定します。

showIcon は、項目に関連するアイコンをユーザー インターフェイスに表示するかどうかを指定します。

separatorBefore は、区切り（たとえば、メニュー項目の水平線またはツールバー ボタンのスペースなど）を、新しい項目の前にメニューに追加するかどうかを指定します。

separatorAfter は、新しい項目の後に区切りをメニューに追加するかどうかを指定します。

offset は、新しい項目の位置を定義する、負ではない整数です。たとえば、0 は新しい項目を最初の項目として追加します（デフォルトでは最後に追加します）。

例

```
<MenuItem action="generateSelect" systemMenu="generate" showIcon="true"/>
```

ツールバー項目

ダイアログまたは出力ウィンドウ向けにカスタムのツールバー項目を定義できます。

書式

```
<ToolbarItem action="action" showLabel="true_false" showIcon="true_false"
   separatorBefore="true_false" separatorAfter="true_false" offset="integer" />
```

ここでの意味は次の通りです。

action（必須）は、Common Objects セクションで定義された操作の識別子で、このツールバー項目で実行される操作を指定します。

showLabel は、項目の表示ラベルをユーザー インターフェイスに表示するかどうかを指定します。

showIcon は、項目に関連するアイコンをユーザー インターフェイスに表示するかどうかを指定します。

separatorBefore は、区切り（たとえば、メニュー項目の水平線またはツールバー ボタンのスペースなど）を、新しい項目の前にメニューに追加するかどうかを指定します。

separatorAfter は、新しい項目の後に区切りをメニューに追加するかどうかを指定します。

offset は、新しい項目の位置を定義する、負ではない整数です。たとえば、0 は新しい項目を最初の項目として追加します（デフォルトでは最後に追加します）。

例

```
<ToolbarItem action="generateDerive" showLabel="true"/>
```

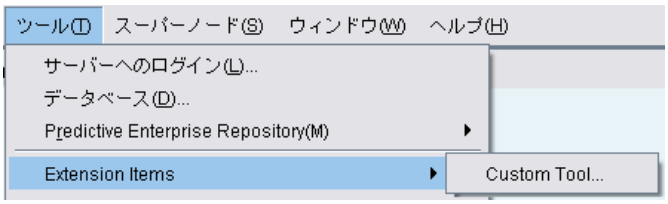
例 :メイン ウィンドウへの追加

これは、メイン ウィンドウの［ツール］メニューの新規項目を追加する特性ファイルの例です。標準オブジェクト（［モデル出力］ウィンドウまたは［ドキュメント出力］ウィンドウなど）は定義されませんが、「メイン ウィンドウの変更」を意味する最上位の Extension に UserInterface 要素が含まれます。その他のすべての UserInterface セクションは、標準オブジェクト定義のいずれかに使用される必要があり、標準オブジェクトと関連するダイアログに影響を与えます。

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0">
   <ExtensionDetails providerTag="example" id="main_window" label="メイン ウィンドウ" version="1.0"
      provider="IBM Corp." copyright="(c) 2005-2006 IBM Corp."
      description="メイン ウィンドウに接続する拡張例"/>
   <Resources/>
   <CommonObjects>
      <Actions>
         <Action id="customTool1" label="カスタム ツール..." labelKey="customTool.LABEL"
            imagePath="images/generate.gif" description="カスタム ツールの起動"
            descriptionKey="customTool.TOOLTIP"/>
         <Action id="customTool2" label="カスタム ツール..." labelKey="customTool.LABEL"
            imagePath="images/generate.gif" description="カスタム ツールの起動"
            descriptionKey="customTool.TOOLTIP"/>
      </Actions>
   </CommonObjects>
   <UserInterface actionHandler="com.spss.cleftest.MainWindowActionHandler">
      <Controls>
         <Menu systemMenu="tools" id ="toolsExtension" separatorBefore="true"
            label="拡張項目 " 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>
```

これにより、[] メニューに新しいサブメニューの［**拡張項目**］が追加されます。この新しいサブメニューには、［**カスタム ツール**］という単一エントリが含まれます。

図 6-4
標準メニューの拡張



XML コードを extension.xml というファイル名で保存し、拡張を CLEF に追加してこの例を使用することができます。 詳細は、 10 章 p.274 CLEF 拡張のテスト を参照してください。

タブ

Tabs セクションでは、以下に表示されるタブを定義します。

- ユーザーが領域上のノードを開いた場合に表示されるダイアログ
- モデル出力ウィンドウ
- ドキュメント出力ウィンドウ

各 Tabs セクションには複数の Tab 要素が含まれている場合があり、それぞれの要素は表示されるカスタム タブを宣言します。

```
<Tabs defaultTab="integer">
   <Tab .../>
   <Tab .../>
   ...
</Tabs>
```

この場合、defaultTab はノードのダイアログまたはウィンドウがストリーム内で初めて開いた場合に表示されるタブを指定する、負ではない整数です。ユーザーが異なるタブを選択する場合、ストリームが有効な間にダイアログまたはウィンドウを閉じて再び開くと、デフォルトのタブではなく、最近表示されたタブが表示されます。タブの番号は 0 から始まります。

その他のタブがダイアログまたはウィンドウに自動的に含まれる場合があります。たとえば、すべてのダイアログおよび出力ウィンドウには［**注釈**］タブが含まれ、すべてのデータ ソースのノードのダイアログには［**フィルタ**］タブおよび［**タイプ**］タブが含まれます。

Tab 要素はタブ ラベル（タブ自体に表示されるテキスト）を宣言し、タブ ラベルが翻訳された場合に参照として使用されるラベル キーも使用する必要があります。

各 Tab 要素内ではパネル設定が行われ、タブのメイン コンテンツ領域がどのように配置されるかを定義します。パネル設定はテキスト ブラウザ、拡張オブジェクト、またはプロパティのいずれかになります。 詳細は、 p. 159 パネル設定 を参照してください。

各 Tab 要素の形式は次のとおりです。

```
<Tab id="identifier" label="display_label" labelKey="label_key" helpLink="help_ID"
      mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" >
   <TextBrowserPanel ... />
   <ExtensionObjectPanel ... />
   <PropertiesPanel ... />
   <ModelViewerPanel ... />
</Tab>
```

ここでの意味は次の通りです。

id は、Java コードのタブを参照するために使用できる一意の識別子です。

label（必須）は、ユーザー インターフェイスに表示される、タブの表示名です。

labelKey は、ローカライズ用のラベルを識別します。

helpLink は、ヘルプ システムを起動する場合に表示されるヘルプ トピックの識別子です。識別子の形式は、ヘルプ システムの種類によって異なります（ヘルプ システムの場所および種類の定義 p. 226 参照）。

HTML Help ：ヘルプ トピックの URL
JavaHelp ：トピック ID

mnemonic は、このコントロールを有効化するために Alt キーと組み合わせて使用するアルファベットの文字です（たとえば、値 S を指定すると、ユーザーは Alt + S キーを使用してこのコントロールを有効化できます）。

mnemonicKey は、ローカライズ用の mnemonic を識別します。mnemonic も mnemonicKey も使用されない場合、mnemonic はこのコントロールに使用できません。 詳細は、 p. 157 アクセス キーとキーボード ショートカット を参照してください。

例

Tab 要素の例については、使用できるさまざまな種類のパネル設定に関する以下の項を参照してください。「テキスト ブラウザ パネル」（ p. 159 ）、「拡張オブジェクト パネル」（ p. 162 ）、「プロパティ パネル」（ p. 164 ）、および 「モデル ビューア パネル」（ p. 166 ）

アクセス キーとキーボード ショートカット

ユーザー インターフェイスの機能に対するマウス アクセスの代わりとして、さまざまなキーの組み合わせを指定して、ユーザーはキーボードを使用して機能にアクセスできます。

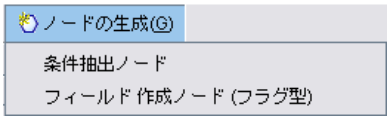
アクセス キー

アクセス キーは、Alt キーと組み合わせて使用できるキーです。メニュー項目、ダイアログ タブ、さまざまなダイアログ コントロールについて、次の要素の mnemonic 属性を使用してアクセス キーを指定できます。

機能	要素	参照...
画面操作（例： メニュー項目）	action	Actions p. 55
メニュー	menu	メニュー p. 150
ダイアログ タブ	tab	タブ p. 155
コントローラ	各種	コントローラ p. 175

たとえば、次のメニューの項目にアクセス キーを指定する手順は次のとおりです。

図 6-5
メニュー項目

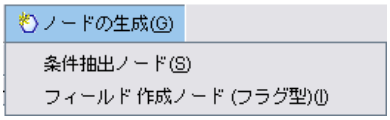


次のコードを使用します。

```
<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="Generates a derive node"
      descriptionKey="generate.deriveNode.TOOLTIP"  mnemonic="D" />
  ...
</Actions>
```

メニュー項目に下線文字が表示されます。

図 6-6
メニュー項目にアクセス キーを使用



ユーザーは、マウスでメニュー項目をクリックするほか、Alt + S キーや Alt + D キーを使用して、メニュー項目にアクセスできます。

ショートカット キー

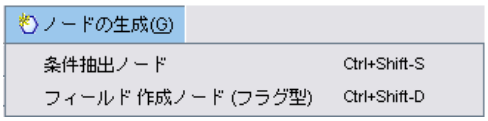
ショートカット キーは、エンド ユーザーがユーザー インターフェイスをナビゲートできるキーの組み合わせです。IBM® SPSS® Modeler は、標準としてさまざまなキーボード ショートカットを提供しています。CLEF では、追加したカスタム メニュー項目にのみショートカットを追加できます。

たとえば、アクセス キーの例で示されたメニューの項目にショートカットを指定するには、次のコードを使用します。

```
<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="Generates a derive node"
      descriptionKey="generate.deriveNode.TOOLTIP"  mnemonic="D"  shortcut="CTRL+SHIFT+D" />
   ...
</Actions>
```

ショートカット キーの組み合わせが、次のようにメニュー項目に追加されます。

図 6-7
メニュー項目にショートカット キーを使用



ユーザーは、マウスでメニュー項目をクリックするほか、キーボード ショートカットを使用して、メニュー項目にアクセスできます。この例のように、アクセス キーと組み合わせてショートカット キーを指定できます。

同じダイアログまたは標準の SPSS Modeler ショートカットにすでに指定されたショートカットを使用しないように注意してください。 詳細は、 A 付録 キーボードのアクセス機能 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

パネル設定

各 Tab 要素には、単一パネルの次のいずれかの設定が含まれます。

テーブル 6-2
パネル設定の種類

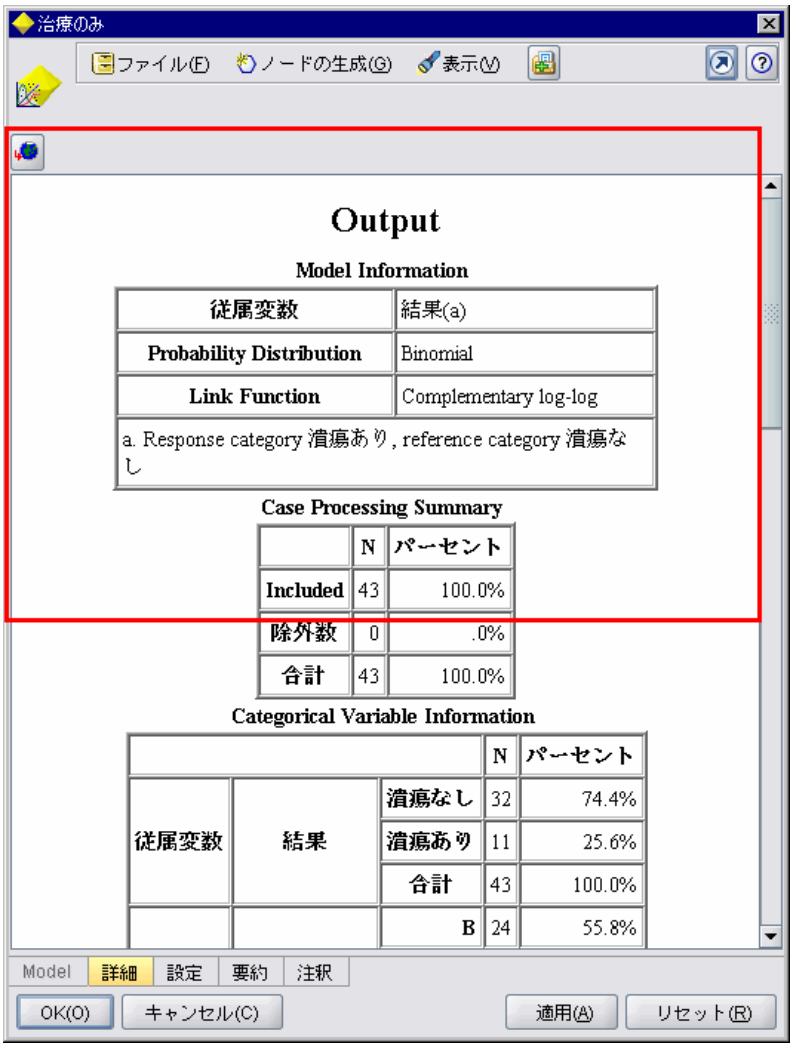
パネル	表示	参照...
テキスト ブラウザ パネル	指定されたコンテナのテキスト コンテンツ。	テキスト ブラウザ パネル p. 159
拡張オブジェクト パネル	指定された Java クラスに定義されたコンテンツ。	拡張オブジェクト パネル p. 162
プロパティ パネル	プロパティ コントロール（ボタン、チェック ボックス、入力フィールド）。	プロパティ パネル p. 164
モデル ビューア パネル	指定されたコンテナからの PMML 形式のモデル出力。	モデル ビューア パネル p. 166

テキスト ブラウザ パネル

テキスト ブラウザ パネルでは、拡張で指定されたコンテナのテキスト コンテンツが表示されます。サポートされた形式（UTF-8 文字コード）は、通常のテキスト形式、HTML 形式、およびリッチ テキスト形式（RTF）です。

以下に、HTML 形式のテキスト ブラウザ パネルを含むモデル出力ウィンドウの例を示します。

図 6-8
テキスト ブラウザ パネルが強調表示されたモデル出力ウィンドウ



書式

```
<TextBrowserPanel container="name" textFormat="text_format" rows="integer"
    columns="integer" wrapLines="true_false" >
    -- advanced custom layout options --
</TextBrowserPanel>
```

ここでの意味は次の通りです。

container（必須）は、パネルの内容を取得するコンテナの名前です。

textFormat（必須）は、パネルに表示されるテキストの形式を指定します。次の 3 つの種類があります。

- plainText
- html
- rtf

rows および columns は、パネルを含むウィンドウが開くと表示されるテキスト行および列の数を指定します。

wrapLines は、長いテキスト行に折り返しを使用する（true）または長いテキスト行を読む場合に水平スクロールを使用する（false）かどうかを指定します。デフォルトは false です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、最初に表示されるテキスト ブラウザ パネルを定義する Tab セクションについて説明します。

```
<Tab label="Advanced" labelKey="advancedTab.LABEL" helpLink="genlin_output_advancedtab.htm">
   <TextBrowserPanel container="advanced_output" textFormat="html" />
</Tab>
```

モデル出力が、タブ設定として同じ ModelOutput セクションに定義されたコンテナに送られます。

```
<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>
```

コンテナは、関連する構築ノードの Execution セクションの CreateDocument 要素で参照されます。

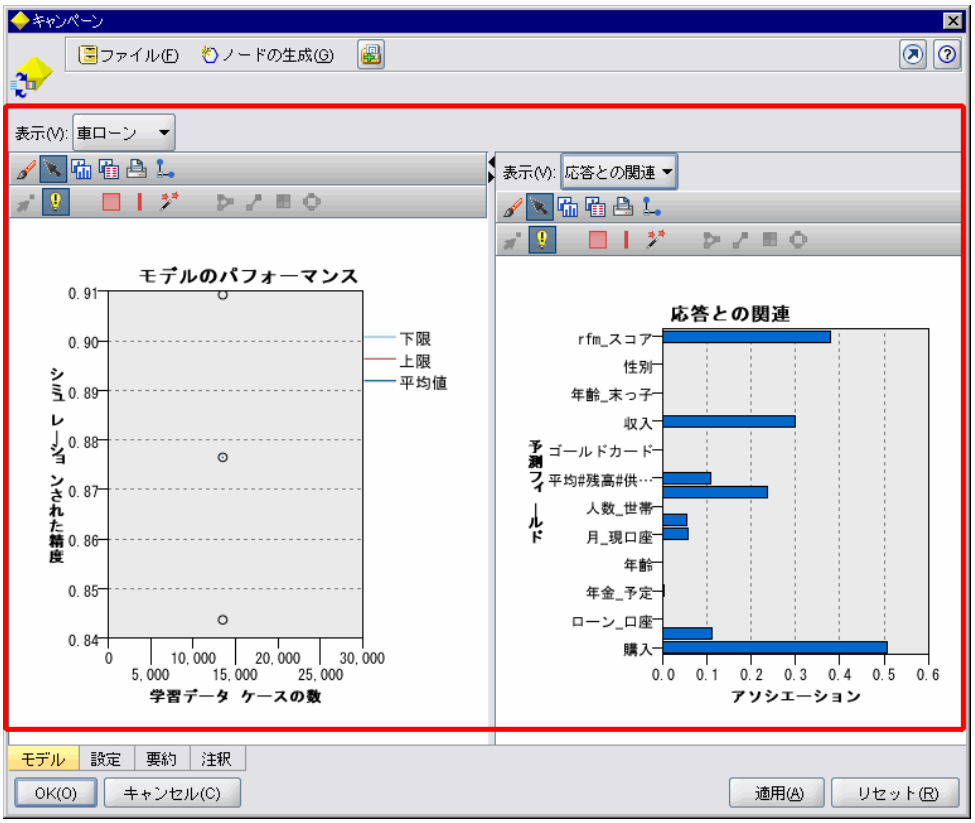
```
<Execution>
 ...
   <Constructors>
      <CreateModelOutput type="generalizedlinear">
         <CreateModel type="generalizedlinear_model" target="model" sourceFile="pmml"/>
         <CreateDocument type="html_output" target="advanced_output" sourceFile="htmloutput"/>
      </CreateModelOutput>
   </Constructors>
</Execution>
```

拡張オブジェクト パネル

拡張オブジェクト パネルはテキスト ブラウザ パネルと同様の方法で動作しますが、コンテナのテキスト コンテンツは表示せず、CLEF Java API で定義された ExtensionObjectPanel インターフェイスを実装する指定された Java クラスのインスタンスを作成します。

以下は、拡張オブジェクト パネルを含むモデル適用ノードのダイアログの例です。

図 6-9
拡張オブジェクト パネルが強調表示されたモデル出力ウィンドウ



書式

```
<ExtensionObjectPanel id="identifier" panelClass="Java_class" >
    -- advanced custom layout options --
</ExtensionObjectPanel>
```

ここでの意味は次の通りです。

id は、Java コードのパネルを参照するために使用できる一意の識別子です。

panelClass（必須）は、パネルが実装する Java クラス名です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、最初に表示される拡張オブジェクト パネルを定義する Tab セクションについて説明します。

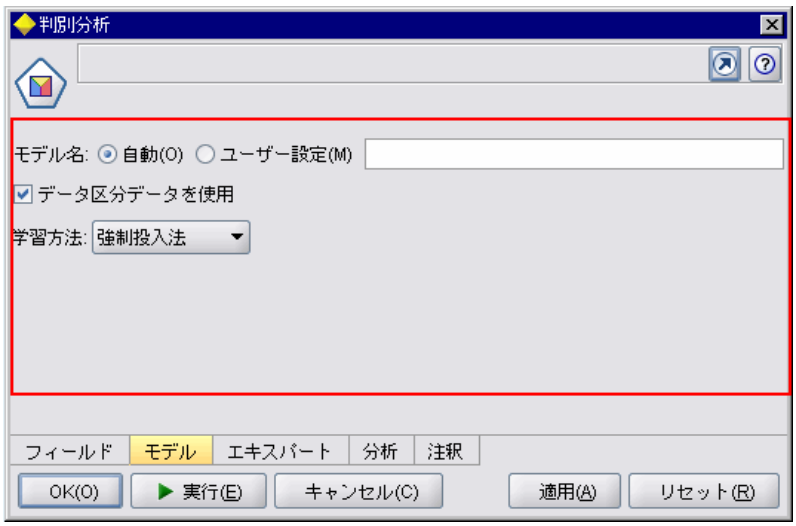
```
<Tab label="モデル" labelKey="Model.LABEL" helpLink="selflearnnode_output.htm">
   <ExtensionObjectPanel id="SelfLearningPanel"
      panelClass="com.spss.clef.selflearning.SelfLearningPanel"/>
</Tab>
```

プロパティ パネル

プロパティ パネルを使用すると、タブまたはプロパティ サブパネル（「プロパティ サブパネル」（ p. 172 ）を参照）に**プロパティ コントロール** を表示することができます。プロパティ コントロールは、画面上に表示されるオブジェクトのプロパティを変更するために使用できる画面コンポーネント（ボタン、チェック ボックス、入力フィールドなど）です。プロパティ パネルでは、ユーザーが [OK] または [適用] をクリックすると、プロパティ コントロールで行われた変更が自動的に適用されます。ユーザーが [キャンセル] または [リセット] をクリックすると、最後に［適用］の操作が行われた以降に行われた変更が破棄されます。

以下に、プロパティ パネルを含むノードのダイアログの例を示します。

図 6-10
プロパティ パネルが強調表示されたノードのダイアログ



次の例では、プロパティ パネルを含むプロパティのサブパネルを示しています。

図 6-11
プロパティ パネルが強調表示されたプロパティ サブパネル

書式

```
<PropertiesPanel id="identifier" label="display_label" labelKey="label_key">
    -- advanced custom layout options --
    -- property control specifications --
</PropertiesPanel>
```

ここでの意味は次の通りです。

id は、Java コードのパネルを参照するために使用できる一意の識別子です。

label プロパティ コントロールのグループに表示する見出しです（最後の例の［**統計**］、［**分類**］および［**ステップワイズ**］など）。

labelKey は、ローカライズ用のラベルを識別します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

個々のプロパティ コントロール設定については、「プロパティ コントロール設定」（ p. 167 ）に記載されています。

例

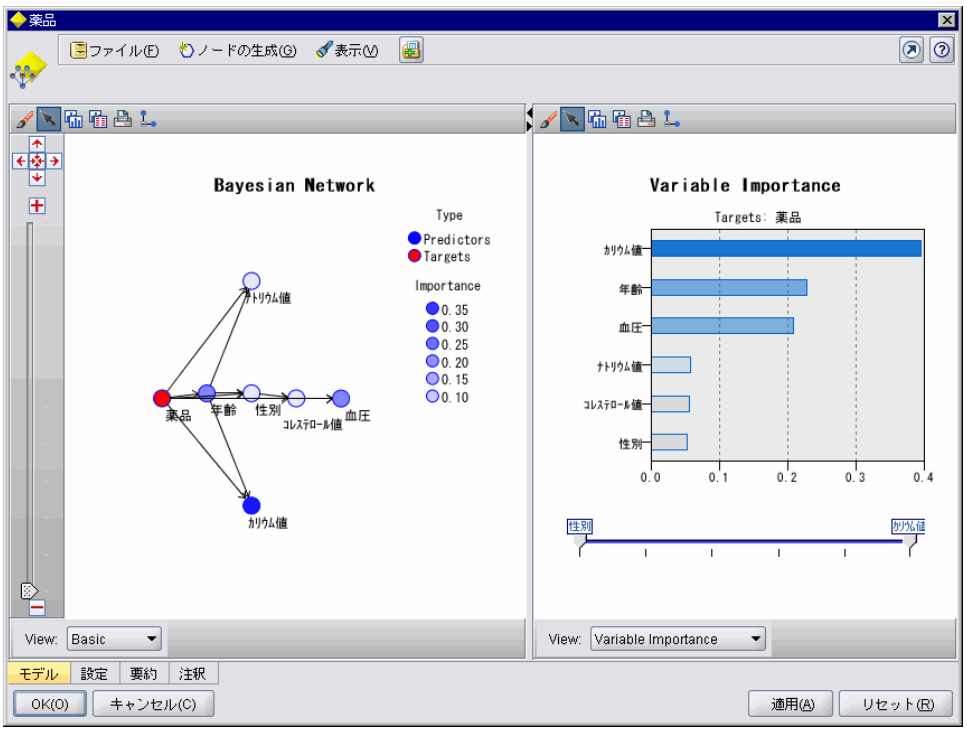
```
<Tab label="モデル" labelKey="Model.LABEL" helpLink="discriminant_node_model.htm">
   <PropertiesPanel>
      <SystemControls controlsId="ModelGeneration" />
      <ComboBoxControl property="method">
         <Layout fill="none" />
      </ComboBoxControl>
   </PropertiesPanel>
</Tab>
```

モデル ビューア パネル

モデル ビューア パネルには、拡張で指定されたコンテナの PMML 形式モデル出力が表示されます。

以下に、モデル ビューア モデルを含むモデル アプライヤ ノード ウィンドウの例を示します。

図 6-12
モデル ビューア パネルが強調表示された出力ウィンドウ



書式

```
<ModelViewerPanel container="container_name">
   -- advanced custom layout options --
</ModelViewerPanel>
```

container（必須）は、モデル出力が割り当てられるコンテナの名前です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

この例では、モデル アプライヤ 設定のモデル ビューア パネルの使用について示しています。モデル出力は、model というコンテナに割り当てられています。以下の例では、モデル アプライヤ設定は model コンテナを配置し、モデル ビューア モデルと関連させます。

```
<Node id="applyBN" type="modelApplier">
   <ModelProvider container="model" isPMML="true" />
   ...
   <Containers>
      <Container name="model" />
   </Containers>
   <UserInterface>
   ...
   <Tabs>
      <Tab label="Model" labelKey="modelTab.LABEL" helpLink="BN_output_modeltab.htm">
         <ModelViewerPanel container="model"/>
      </Tab>
   ...
   </Tabs>
   </UserInterface>
   ...
</Node>
```

プロパティ コントロール設定

プロパティ コントロールは、画面上に表示されるオブジェクトのプロパティを変更するために使用されるボタン、チェック ボックス、入力フィールドなどの画面コンポーネントです。プロパティ コントロール設定の形式は、次のようなプロパティ コントロールの種類によって異なります。

- UI コンポーネント
- プロパティ パネル
- コントローラ

UI コンポーネント コントロールは、操作ボタン、画面上の静的テキスト、システム コントロール（すべてのダイアログに共通するプロパティを操作する一連のコントロール）です。

プロパティ パネル コントロールは、プロパティ パネル設定内の個々のパネルです。

コントローラは、プロパティ コントロールの最大グループを形成し、チェック ボックス、選択情報、スピン コントロールなどの項目が含まれます。

UI コンポーネント コントロール

UI コンポーネント コントロールは、以下のとおりです。

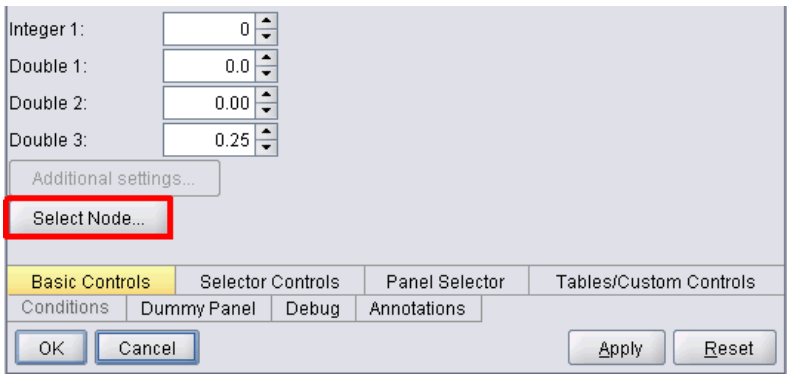
テーブル 6-3
UI コンポーネント コントロール

コントロール	説明
ActionButton	クリックすると事前設定された操作を実行する画面上のボタン。
StaticText	画面上に表示される非可変のテキスト文字列。
SystemControls	すべてのモデルに共通するプロパティを操作するコントロールの標準セット。

操作ボタン

Common Objects セクションで指定された操作を実行するダイアログまたはツールバー ボタンを定義します。ユーザーがこのボタンをクリックすると、新しい画面を表示するなどの操作が実行されます。

図 6-13
操作ボタンが強調表示されたダイアログ



書式

```
<ActionButton action="action" showLabel="true_false" showIcon="true_false" >
   -- advanced custom layout options --
</ActionButton>
```

ここでの意味は次の通りです。

action（必須）は、実行される操作の識別子です。

showLabel は、ボタン ラベルを表示する（true）かまたは非表示にする（false）かを指定します（たとえば、ツールバー ボタンの場合、ラベルではなくアイコンの表示を選択します）。デフォルトは true です。

showIcon はボタンに関連するアイコンを表示する（true）か、または非表示にする（false）かを選択します。デフォルトは false です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

最初に表示される操作ボタンを作成するコードは、次のとおりです。

```
<ActionButton action="generateSelect"/>
```

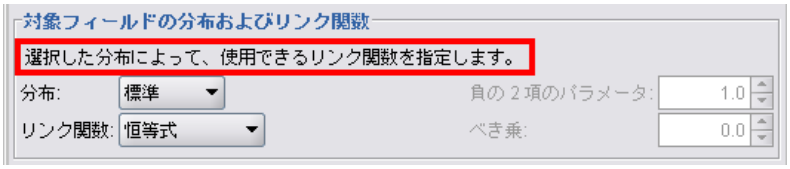
Common Objects セクションで定義される操作は次のとおりです（ボタン ラベルもここで定義されます）。

```
<CommonObjects extensionListenerClass="com.spss.cleftest.TestExtensionListener">
   ...
  <Actions>
   <Action id="generateSelect" label="条件抽出ノード..." labelKey="generate.selectNode.LABEL"
      imagePath="images/generate.gif" description="Generates a select node"
      descriptionKey="generate.selectNode.TOOLTIP"/>
    ...
  </Actions>
</CommonObjects>
```

静的テキスト

この要素を使用して、ダイアログまたは出力ウィンドウに非可変のテキスト文字列を使用することができます。以下の例では、静的テキストを使用するプロパティ パネルを示しています。

図 6-14
静的テキストを強調表示したプロパティ パネル



書式

```
<StaticText text="static_text" textKey="text_key" >
   -- advanced custom layout options --
</StaticText>
```

ここでの意味は次の通りです。

text は、使用するテキスト文字列です。

textKey は、ローカライズ用の静的テキストを識別します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

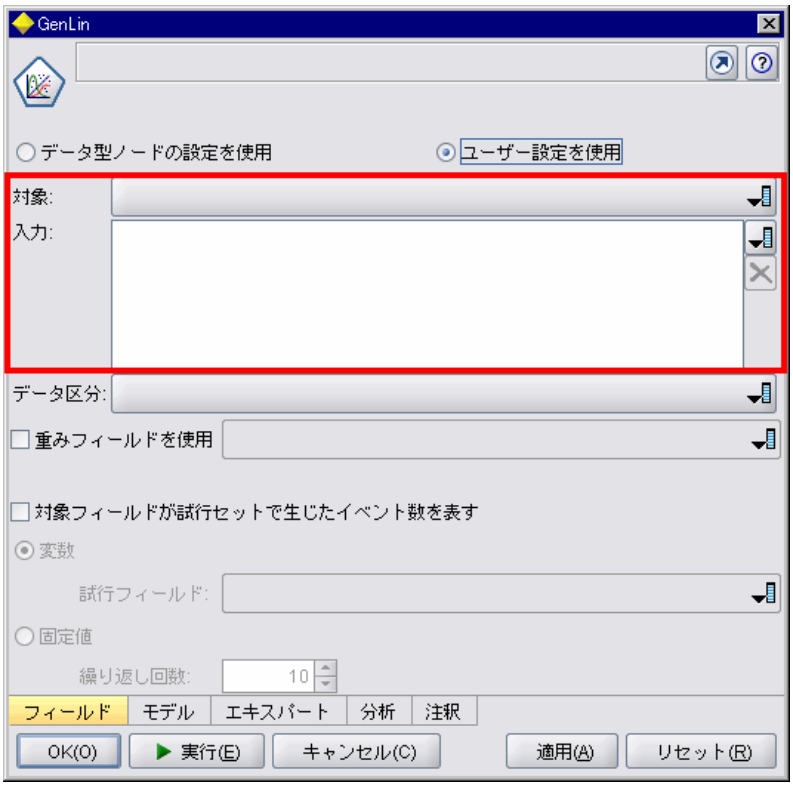
以下の例では、最初に表示される静的テキストに使用される宣言を示します。

```
<StaticText text="The distribution that you choose determines which link functions are available."
   textKey="Genlin_staticText1"/>
```

システム コントロール

一部のプロパティは、すべてのモデルに共通です。モデル ビルダー ノードでは、システム コントロールは共通するプロパティを操作するコントロールの標準セットです。

図 6-15
システム コントロールが強調表示されたダイアログの例



書式

```
<SystemControls controlsID="identifier" >
    -- advanced custom layout options --
</SystemControls>
```

controlsID は、コントロールのセットの識別子です。この識別子は、モデル ビルダーの宣言の ModelingFields 要素の controlsID 属性に指定されている識別子と同じものである必要があります（「モデル ビルダー」（ p. 70 ） を参照）。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

この例では、前述の例のシステム コントロールに使用された宣言を示しています。

ノード設定の Model Builder セクションでは、次で一連のシステム コントロールのセットが定義され、この場合モデルの入力フィールドおよび出力フィールドのフィールド ピッカーを保持します。

```
<ModelBuilder ... >
   <ModelingFields controlsId="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>
```

その後、このセットのコントロールはそれらが表示されるモデル ビルダー ノードのダイアログのタブ定義で参照されます。

```
<Tab label="Fields" labelKey="Fields.LABEL" helpLink="genlin_node_fieldstab.htm">
   <PropertiesPanel>
      <SystemControls controlsId="modelingFields">
      </SystemControls>
    ...
   </PropertiesPanel>
</Tab>
```

プロパティ パネル コントロール

プロパティ パネル コントロールは次のとおりです。

テーブル 6-4
プロパティ パネル コントロール

コントロール	説明
PropertiesSubPanel	ユーザーがプロパティ パネルのボタンをクリックすると表示される各ダイアログ。
PropertiesPanel	プロパティ サブパネルの宣言にネストされた、または最上位のプロパティ パネルの宣言にネストされたプロパティ。

プロパティ サブパネル

ユーザーがプロパティ パネルのボタンをクリックすると表示される各ダイアログを定義します。プロパティ サブパネルの宣言は、タブのメインプロパティ パネル設定の一部として作成されます。

書式

```
<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>
```

ここでの意味は次の通りです。

buttonLabel は、サブパネルへのアクセスを提供するボタンのラベルです。

buttonLabelKey ローカライズ用のボタンのラベルを識別します。

dialogTitle は、サブパネル ダイアログのタイトル バーに表示されるテキストです。

dialogTitleKey は、ローカライズ用のサブパネル ダイアログのタイトルを識別します。

helpLink は、ヘルプ システムを起動する場合に表示されるヘルプ トピックの識別子です。識別子の形式は、ヘルプ システムの種類によって異なります（ヘルプ システムの場所および種類の定義 p. 226 参照）。

HTML Help ：ヘルプ トピックの URL
JavaHelp ：トピック ID

mnemonic は、このコントロールを有効化するために Alt キーと組み合わせて使用するアルファベットの文字です（たとえば、値 S を指定すると、ユーザーは Alt + S キーを使用してこのコントロールを有効化できます）。

mnemonicKey は、ローカライズ用の mnemonic を識別します。mnemonic も mnemonicKey も使用されない場合、mnemonic はこのコントロールに使用できません。 詳細は、 p. 157 アクセス キーとキーボード ショートカット を参照してください。

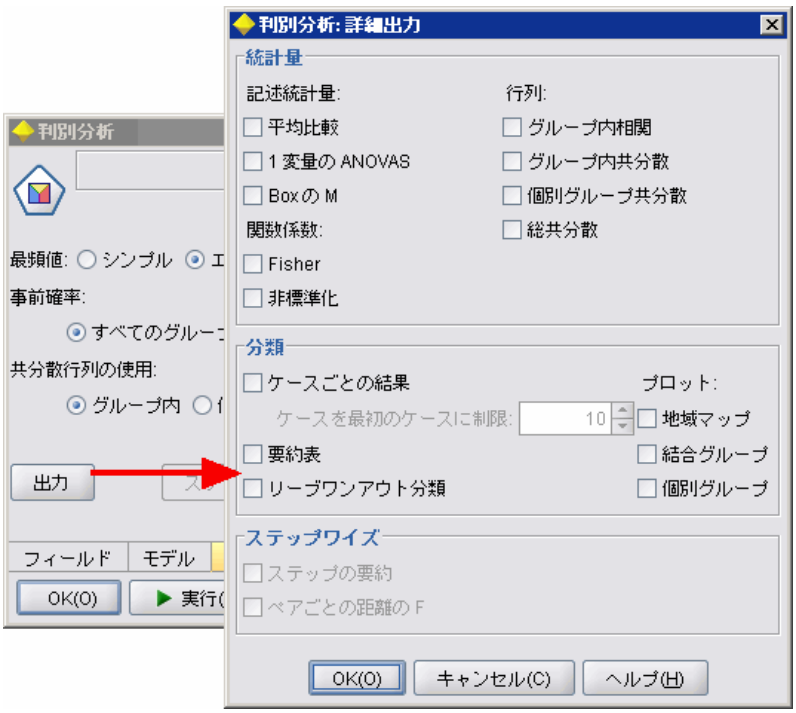
詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、 p. 207 詳細なカスタム レイアウト を参照してください。

個々のプロパティ コントロール設定については、「プロパティ コントロール設定」（ p. 167 ）に記載されています。

例

以下に、ユーザーがタブのメイン プロパティ パネルの［出力］ボタンをクリックした場合に表示されるプロパティ サブパネルを示します。

図 6-16
プロパティ サブパネル



以下のコードは、上記のプロパティ サブパネルを取得するために使用する宣言の大部分を示します。サブパネルの宣言内で、各フィールド グループ（**統計**、**分類**および**ステップワイズ**）は、独自のプロパティ パネル設定を保持します。

```
<PropertiesSubPanel buttonLabel="出力..." buttonLabelKey="OutputSubPanel.LABEL"
   dialogTitle="判別分析:詳細出力" dialogTitleKey="AdvancedOutputSubDialog.LABEL"
   helpLink="discriminant_node_outputdlg.htm">
   ...
   <PropertiesPanel>
      <PropertiesPanel label="統計" ...>
       ...
      </PropertiesPanel>
      <PropertiesPanel label="分類" ...>
       ...
      </PropertiesPanel>
      <PropertiesPanel label="ステップワイズ" ...>
       ...
```

```
      </PropertiesPanel>
   </PropertiesPanel>
</PropertiesSubPanel>
```

プロパティ パネル（ネスト）

プロパティ サブパネルの宣言内にプロパティ パネル設定をネストし、サブパネルから表示されるダイアログの内容を定義することができます。 詳細は、 p. 172 プロパティ サブパネル を参照してください。

また、最上位のプロパティ パネルの宣言内にプロパティ パネル設定をネストすることもできます。複数のプロパティ パネルで構成されたタブ全体の内容を、タブの特定の特定のボタンが選択されているかどうかによって有効化または無効化する場合の例を示します。この場合、タブ設定は次のようになります。

```
<Tab .... >
   <PropertiesPanel>
    --- button specification ---
      <PropertiesPanel>
         <Enabled>
          --- condition involving button value ---
         </Enabled>
       ...
      </PropertiesPanel>
      <PropertiesPanel>
         <Enabled>
          --- condition involving button value ---
         </Enabled>
       ...
      </PropertiesPanel>
       ...
   </PropertiesPanel>
</Tab>
```

ネストされたプロパティ パネル設定の形式は、最上位の要素の形式の同じです。 詳細は、 p. 164 プロパティ パネル を参照してください。

コントローラ

コントローラは、プロパティ コントロール最も大きいグループを形成します。

テーブル 6-5
コントローラ

コントロール	説明
CheckBoxControl	チェック ボックス。
CheckBoxGroupControl	チェック ボックスのセット。enum 値ごとに 1 つのチェック ボックス。
ClientDirectoryChooserControl	ユーザーがクライアントのディレクトリを選択できる単一行のテキスト フィールドおよび関連ボタン。
ClientFileChooserControl	ユーザーがクライアントのファイルを選択できる単一行のテキスト フィールドおよび関連ボタン。
ComboBoxControl	enum 値を含む選択情報のドロップダウン リスト。
DBConnectionChooserControl	ユーザーはデータ ソースおよびデータベースへの接続を選択可能。
DBTableChooserControl	データベース接続後、ユーザーはデータベース テーブルを選択可能。
MultiFieldChooserControl	(ノードのみ) ユーザーがリストからフィールドを選択できる、フィールド名のリスト。
MultiItemChooserControl	値のリストから、1 つまたは複数の項目を選択できます。
PasswordBoxControl	入力文字が非表示になる単一行のテキスト フィールド。
PropertyControl	ユーザー定義可能なプロパティのコントロール。
RadioButtonGroupControl	1 回に 1 つのボタンのみ選択できるラジオ ボタンのセット。enum プロパティの場合、 enum 値ごとに 1 つのラジオ ボタンが表示されます。ブール プロパティの場合、2 つのラジオ ボタンが表示されます。
ServerDirectoryChooserControl	ユーザーがサーバーのディレクトリを選択できる単一行のテキスト フィールドおよび関連ボタン。
ServerFileChooserControl	ユーザーがサーバーのファイルを選択できる単一行のテキスト フィールドおよび関連ボタン。
SingleFieldChooserControl	(ノードのみ) ユーザーがリストから 1 つのフィールドを選択できる、フィールド名のリスト。
SingleItemChooserControl	値のリストから、1 つの項目を選択できます。
SpinnerControl	スピン コントロール (値を変更するための、上下の矢印の付いた数値型フィールド)。
TableControl	ダイアログまたはウィンドウにテーブルを追加。
TextAreaControl	複数行のテキストフィールド。
TextBoxControl	単一行のテキストフィールド。

コントローラの属性

コントローラの設定には、次の属性が含まれます。

property="value" showLabel="true_false" label="display_label" labelKey="label_key" labelWidth="label_width" labelAbove="true_false" description="description" descriptionKey="description_key" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"

ここでの意味は次の通りです。

property（必須）は、プロパティ コントロールの一意の識別子です。

showLabel はボタンに関連するプロパティ コントロールの表示ラベルを表示する（true）か、または非表示にする（false）かを指定します。デフォルトは true です。

label は、ユーザー インターフェイスに表示される、プロパティ コントロールの表示名です。また、この値はプロパティ コントロールのアクセス可能な簡単な説明としても動作します。 詳細は、 8 章 p. 238 アクセシビリティ を参照してください。

labelKey は、ローカライズ用のラベルを識別します。

labelWidth は、ラベルが広がる表示グリッド列の数値です。デフォルトは 1 です。

labelAbove は、コントロールのラベルを上部に表示する（true）またはコントロールに隣接して表示する（false）かを指定します。デフォルトは false です。

description は、カーソルをコントロールに移動した場合に表示される、ツールヒントのテキストです。また、この値はプロパティ コントロールのアクセス可能な詳細な説明としても動作します。 詳細は、 8 章 p. 238 アクセシビリティ を参照してください。

descriptionKey は、ローカライズ用の説明を識別します。

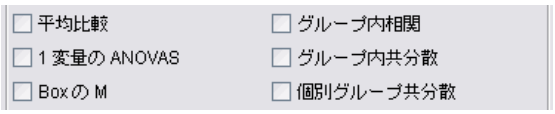
mnemonic は、このコントロールを有効化するために Alt キーと組み合わせて使用するアルファベットの文字です（たとえば、値 S を指定すると、ユーザーは Alt + S キーを使用してこのコントロールを有効化できます）。

mnemonicKey は、ローカライズ用の mnemonic を識別します。mnemonic も mnemonicKey も使用されない場合、mnemonic はこのコントロールに使用できません。 詳細は、 p. 157 アクセス キーとキーボード ショートカット を参照してください。

チェック ボックス コントロール

チェック ボックスを定義します。

図 6-17
チェック ボックス



書式

```
<CheckBoxControl controller_attributes invert="true_false" >
   -- advanced custom layout options --
</CheckBoxControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

invert はあまり使用されることはありませんが、true に設定された場合、チェック ボックスの選択および選択解除の影響を逆にします。デフォルトは false です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、最初に表示されるチェック ボックスを配置するために使用されるコードを示します（チェック ボックスのラベルは、特性ファイル内の別の場所で定義されます）。Layout 要素については、「詳細なカスタム レイアウト 」（ p. 207 ）に記載されています。

```
<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>
```

チェック ボックス グループ コントロール

グループ化され、単一ユニットとして扱われるチェック ボックスのセットを定義します。グループのメンバーを定義する列挙リストのプロパティと組み合わせた場合にのみ使用できます。

図 6-18
チェック ボックス グループ

書式

```
<CheckBoxGroupControl controller_attributes rows="integer" layoutByRow="true_false"
   useSubPanel="true_false" >
   -- advanced custom layout options --
</CheckBoxGroupControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

rows は正の整数で、チェック ボックス グループは表示される画面の行数を指定します。デフォルトは 1 です。

layoutByRow は、チェック ボックスを最初は行に沿って配置する（true）か、列に沿って配置する（false）かを指定します。デフォルトは true です。ラジオ ボタン グループの layoutByRow の同様の使用方法については、「コントロールの順序の変更」（ p. 207 ）を参照してください。

useSubPanel は、チェック ボックスをサブパネルとして配置する（true）か、配置しない（false）かを指定します。デフォルトは true です。

チェック ボックス グループは通常、グループのすべてのボックスを含むサブパネルとして配置されます。ただし、チェック ボックス グループが隣接するテキスト フィールドと関連する場合、整列の問題が生じる場合があります。useSubPanel を false に設定すると、この問題を解決することができます。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

最初に表示されるチェック ボックス グループを作成するコードは、次のとおりです。

```
<CheckBoxGroupControl property="enum3" label="列挙 3" labelKey="enum3.LABEL"/>
```

個々のチェック ボックスに関連するラベルおよび値は、関連するノードの Properties セクションで次のように定義されます。

```
<Property name="enum3" valueType="enum" isList="true" defaultValue="[value1 value3]">
   <Enumeration>
      <Enum value="value1" label="値 3.1" labelKey="enum3.value1.LABEL"/>
      <Enum value="value2" label="値 3.2" labelKey="enum3.value2.LABEL"/>
      <Enum value="value3" label="値 3.3" labelKey="enum3.value3.LABEL"/>
      <Enum value="value4" label="値 3.4" labelKey="enum3.value4.LABEL"/>
      <Enum value="value5" label="値 3.5" labelKey="enum3.value5.LABEL"/>
   </Enumeration>
</Property>
```

クライアント ディレクトリ設定コントロール

ユーザーがクライアントのディレクトリを選択できる単一行のテキスト フィールドおよび関連ボタンを定義します。ディレクトリは、常に存在する必要があります。ユーザーは、モード設定に応じてファイルをディレクトリから開くか、ファイルをディレクトリに保存することができます。

図 6-19
クライアント ディレクトリ設定コントロール

ユーザーは、テキスト フィールドにディレクトリ パスまたはディレクトリ名を直接入力するか、隣接するボタンをクリックしてディレクトリを選択できるダイアログを表示することができます。

書式

```
<ClientDirectoryChooserControl controller_attributes mode="chooser_mode" >
   -- advanced custom layout options --
</ClientDirectoryChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

mode は、ユーザーがディレクトリを選択する、ダイアログに表示されるボタンを定義します。ボタンは次のいずれかです。

- open（デフォルト）は、[開く] ボタンを表示します。
- save は、[保存] ボタンを表示します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

```
<ClientDirectoryChooserControl property="directory2" label="クライアント ディレクトリ"
   labelKey="directory2.LABEL"/>
```

クライアント ファイル設定コントロール

ユーザーがクライアントのファイルを選択できる単一行のテキスト フィールドおよび関連ボタンを定義します。ディレクトリは、すでに存在する必要があります。ユーザーはノード設定によって、ファイルを開くか保存することができます。

図 6-20
クライアント ファイル設定コントロール

ユーザーは、テキスト フィールドにファイル パスまたはディレクトリ名を直接入力するか、隣接するボタンをクリックしてファイルを選択できるダイアログを表示することができます。

書式

```
<ClientFileChooserControl controller_attributes mode="chooser_mode" >
   -- advanced custom layout options --
</ClientFileChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

mode は、ユーザーがファイルを選択する、ダイアログに表示されるボタンを定義します。ボタンは次のいずれかです。

- open（デフォルト）は、［開く］ボタンを表示します。
- save は、［保存］ボタンを表示します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

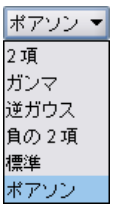
例

```
<ClientFileChooserControl property="file2" label="クライアント ファイル" labelKey="file2.LABEL"/>
```

選択情報コントロール

選択情報のドロップダウン リストを定義します。

図 6-21
選択情報



書式

```
<ComboBoxControl controller_attributes >
   -- advanced custom layout options --
</ComboBoxControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、前述の説明に表示した選択情報のドロップダウン リストを配置するために使用するコードを示します。

```
<ComboBoxControl property="distribution" >
   <Layout rowIncrement="0" gridWidth="1" fill="none"/>
</ComboBoxControl>
```

Layout 要素については、「詳細なカスタム レイアウト 」（ p. 207 ）に記載されています。

注 ： 実際のリストのエントリは、関連ノードの Properties セクションで定義されます。この場合、distribution プロパティの宣言内で列挙リストとして定義されます。

```
<Property name="distribution" valueType="enum" label="分布" labelKey="distribution.LABEL"
defaultValue="NORMAL">
   <Enumeration>
      <Enum value="BINOMIAL" label="2 項" labelKey="distribution.BINOMIAL.LABEL"/>
      <Enum value="GAMMA" label="ガンマ" labelKey="distribution.GAMMA.LABEL"/>
      <Enum value="IGAUSS" label="逆ガウス" labelKey="distribution.IGAUSS.LABEL"/>
      <Enum value="NEGBIN" label="負の 2 項" labelKey="distribution.NEGBIN.LABEL"/>
      <Enum value="NORMAL" label="正規" labelKey="distribution.NORMAL.LABEL"/>
      <Enum value="POISSON" label="ポワソン" labelKey="distribution.POISSON.LABEL"/>
   </Enumeration>
</Property>
```

データベース接続設定コントロール

ユーザーがデータ ソースおよびデータベースへの接続を選択できるコントロールを定義します。

図 6-22
データベース接続設定コントロール

ユーザーはテキストをテキスト フィールドに入力することはできませんが、ボタンをクリックして標準 IBM® SPSS® Modeler の［データベース接続］ダイアログを表示する必要があります。

図 6-23
[データベース接続] ダイアログ

正常に接続されると、接続の詳細情報が、データベース接続設定コントロールのテキスト フィールドに表示されます。

書式

```
<DBConnectionChooserControl controller_attributes >
   -- advanced custom layout options --
</DBConnectionChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、コントロールがどのように接続文字列に使用できる文字列プロパティの定義を要求するかを示します。

```
<Node ... >
   <Properties>
    ...
      <Property name="dbconnect" valueType="databaseConnection" />
   </Properties>
    ...
```

```
    <UserInterface>
    ...
      <Tabs>
        <Tab label="データベース">
          <PropertiesPanel>
            <DBConnectionChooserControl property="dbconnect" label="データベース接続"/>
          ...
          </PropertiesPanel>
          ...
    </UserInterface>
</Node>
```

データベース テーブル設定コントロール

データベース接続後、ユーザーがデータベース テーブルを選択できるテキスト フィールドおよび関連するボタンを定義します。

図 6-24
データベース テーブル設定コントロール

ユーザーはテキスト フィールドに直接テーブル名を入力するか、ボタンをクリックして次のようなリストからテーブル名を選択することができます。

図 6-25
データベース テーブル リスト

書式

```
<DBTableChooserControl connectionProperty="DB_connection_property" controller_attributes >
   -- advanced custom layout options --
</DBTableChooserControl>
```

ここでの意味は次の通りです。

connectionProperty は、すでに定義されているデータベース接続プロパティの名前です。これは、ノードに以前定義されている DBConnectionChooserControl 要素の property 属性の値です。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

この例は DBConnectionChooserControl の例に従い、データベース接続確率後どのように DBTableChooserControl 要素を使用してデータベース テーブルを選択するかを示します。

```
<Node ... >
   <Properties>
    ...
      <Property name="dbconnect" valueType="databaseConnection" />
      <Property name="dbtable" valueType="string" />
   </Properties>
    ...
   <UserInterface>
    ...
      <Tabs>
         <Tab label="データベース">
            <PropertiesPanel>
               <DBConnectionChooserControl property="dbconnect" label="データベース接続"/>
               <DBTableChooserControl property="dbtable" connectionProperty="dbconnect"
                  label="データベース テーブル" />
             ...
            </PropertiesPanel>
    ...
   </UserInterface>
</Node>
```

複数フィールド設定コントロール

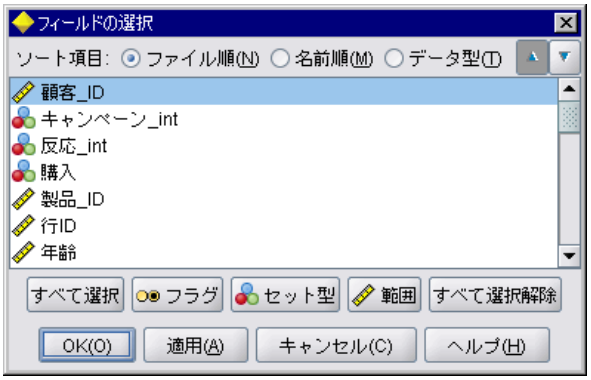
ユーザーがリストから 1 つまたは複数のフィールドを選択できるコントロールを定義します。

図 6-26
複数フィールド設定コントロール

ユーザーがこのコントロールをクリックすると、ユーザーが 1 つまたは複数のフィールドを選択できるフィールドのリストが表示されます。

セットは、このノードで表示されるすべてのフィールドで構成されています。フィールドがこのノードの上流で詳細にフィルタリングされている場合、フィルタを通過したフィールドのみが表示されます。特定のストレージ タイプおよびデータ型のフィールドのみが選択できるよう指定することによって、リストをより詳細に制限することができます。

図 6-27
複数フィールドのリスト



複数フィールド設定コントロールはプロパティ属性を指定します。プロパティ属性はファイル内の別の場所で宣言され、ノードのダイアログにリストがどのように表示されるかを定義します。

書式

```
<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>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

2 つの別の属性を指定してフィールドのリストをさらに制限することもできます。2 つの属性のいずれかは、次のリストから選択する必要があります。

- storage は、リスト内に表示されるフィールドのストレージ タイプを指定するリストのプロパティです。たとえば、storage="[integer real]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意

味します。可能なストレージ タイプのセットについては、データおよびストレージ タイプ p. 251 のテーブルを参照してください。

- onlyNumeric が true に設定されている場合、数値型ストレージのフィールドのみが一覧表示されます。
- onlySymbolic が true に設定されている場合、シンボル値のストレージ タイプのフィールド（文字列）のみが一覧表示されます。
- onlyDatetime が true に設定されている場合、日付と時間のストレージ タイプのみが一覧表示されます。

2 番目の属性を以下から選択する必要があります。

- types は、リスト内に表示されるフィールドのデータ型を指定するリストのプロパティです。たとえば、types="[range flag]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。使用できるデータ 型のセットは次のとおりです。

  range

  flag

  set

  orderedSet

  numeric

  discrete

  typeless
- onlyRanges が true に設定されている場合、範囲のデータ型のフィールドのみが一覧表示されます。
- onlyDiscrete が true に設定されている場合、不連続な（フラグ、セット、データ型不明）のデータ型のみが一覧表示されます。

たとえば、storage="[integer]" および types="[flag]" を指定するコントロールでは、フラグ型である整数フィールドのみがリスト内に表示されます。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

注 ： このコントロールは、Node 要素の定義にのみ使用されます。出力データモデルの定義内で複数フィールド設定を指定するには、次の形式を使用します。

```
<OutputDataModel mode="mode">
...
  <ForEach var="field" inProperty="prop_name">
    <AddField name="${field_name}_NEW" fieldRef="${field_name}" />
  </ForEach>
```

```
...
</OutputDataModel>
```

詳細は、 4 章 p.81 出力データ モデル を参照してください。ForEach 要素については、「ForEach 要素による反復」（ p.92 ）に記載されています。AddField については、「フィールドの追加」（ p.89 ）に記載されています。

例

次の例では、前述の例の複数フィールド設定コントロールを指定するために使用するコードを示します。

```
<MultiFieldChooserControl property="inputs" >
   <Enabled>
      <Condition control="custom_fields" op="equals" value="true"/>
   </Enabled>
</MultiFieldChooserControl>
```

Enabled セクションによって、custom_fields コントロールが選択された場合にのみコントロールを有効化できます。

注 ： このリストの内容は、関連するノードの Properties セクション内の inputs プロパティの宣言に支配されます。

```
<Property name="inputs" valueType="string" isList="true" label="入力" labelKey="inputs.LABEL"/>
```

複数項目設定コントロール

ユーザーが値のリストから 1 つまたは複数の項目を選択できるコントロールを定義します。プロパティを、値のリストを持つカタログと関連付けます。 詳細は、 4 章 p.56 カタログ を参照してください。

図 6-28
複数項目設定コントロール

書式

```
<MultiItemChooserControl controller_attributes catalog="catalog_name" >
   -- advanced custom layout options --
</MultiItemChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

catalog（必須）は、関連するカタログの名前です。カタログが取得されるライブラリは、Execution セクションの Module 要素で指定されているライブラリです。 詳細は、 4 章 p. 78 モジュール を参照してください。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、 p. 207 詳細なカスタム レイアウト を参照してください。

例

```
<MultiItemChooserControl property="selection2" catalog="cat2" />
```

property 属性で参照されるプロパティ（この場合は selection2）は、isList="true" 属性を持つプロパティである必要があります。MultiItemChooserControl の使用についての説明および例は、「カタログ」（ p. 56 ）を参照してください。

パスワード ボックス コントロール

入力文字が入力されると非表示になる、単一行のテキスト フィールドを定義します。

図 6-29
パスワード ボックス コントロール

書式

```
<PasswordBoxControl controller_attributes columns="integer" >
   -- advanced custom layout options --
</PasswordBoxControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

columns は正の整数で、パスワード ボックスが占める文字の列数を定義します。デフォルトは 20 です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、 p. 207 詳細なカスタム レイアウト を参照してください。

例

```
<PasswordBoxControl property="encrypted_string1" label="暗号化文字列
1" labelKey="encryptedString1.LABEL"/>
```

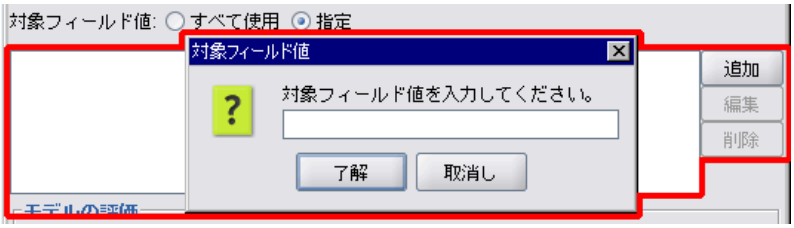
テキスト フィールドは、関連するノードの Properties セクション内で暗号化された文字列として定義されているプロパティと関連させて暗号化します。

```
<Property name="encrypted_string1" valueType="encryptedString"/>
```

プロパティ コントロール

プロパティ コントロールは、ユーザーがノードのプロパティを入力できる、完全にユーザー定義可能なコントロールです。プロセスは、ユーザーによって記述された Java クラスによって処理されます。プロパティ コントロールの例は次のとおりです。

図 6-30
プロパティ コントロールの例を強調表示したダイアログのセクション



書式

```
<PropertyControl controller_attributes controlClass="Java_class" >
   -- advanced custom layout options --
</PropertyControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

controlClass（必須）は、プロパティ コントロールを実装する Java クラスへの .jar ファイル内のパスです。（注 : .jar ファイルは、Resources セクションの JarFile 要素で宣言されます。詳細は、 4 章 p. 48 Jar ファイル を参照してください。）

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

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>
```

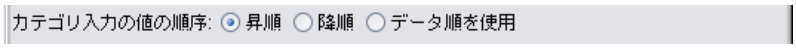
プロパティ コントロールは、関連ノードの Properties セクションで定義されたプロパティと関連します。

```
<Property name="target_field_values_specify" valueType="string" isList="true" label=""
    labelKey="target_field_values_specify.LABEL"/>
```

ラジオ ボタン グループ コントロール

1 回に 1 つのボタンのみ選択できるラジオ ボタンのセットを定義します。

図 6-31
ラジオ ボタン グループ コントロール



ラジオ ボタン グループ コントロールには、それぞれグループを特定のプロパティと関連付けるプロパティ属性があります。このプロパティは、ファイル内の別の場所で定義され、グループを構成するボタンを指定します。

関連するプロパティは、列挙リストまたはブール プロパティとなります。列挙リスト（プロパティ属性が valueType="enum"）の場合、各 enum 値にラジオ ボタンが 1 つずつ表示されます。ブール プロパティ（where valueType="boolean"）の場合、2 つのボタンが常に表示されます。

書式

```
<RadioButtonGroupControl controller_attributes
    rows="integer" 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>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

rows は正の整数で、グループが表示される画面の行数を指定します。デフォルトは 1 です。

layoutByRow は、ラジオ ボタンを最初は行に沿って配置する（true）か、列に沿って配置する（false）かを指定します。デフォルトは true です。ラジオ ボタン グループの layoutByRow の同様の使用例については、「コントロールの順序の変更」（ p. 207 ）を参照してください。

useSubPanel は、ラジオ ボタンをサブパネルとして配置する（true）か、配置しない（false）かを指定します。デフォルトは true です。

ラジオ ボタン グループは通常、グループのすべてのボタンを含むサブパネルとして配置されます。ただし、ラジオ ボタン グループが隣接するテキスト フィールドと関連する場合、整列の問題が生じる場合があります。useSubPanel を false に設定すると、この問題を解決することができます。

falseLabel は、ブール プロパティの偽（false）の値のラベルです（下の 2 番目の例を参照）。ブール プロパティでのみ使用され、その場合は必須です。

falseLabelKey は、ローカライズ用のラベルを識別します。

trueLabel は、ブール プロパティの真（true）の値のラベルです（下の 2 番目の例を参照）。ブール プロパティでのみ使用され、その場合は必須です。

trueLabelKey ローカライズ用の真（true）のラベルを識別します。

true に設定されている場合、trueFirst はブール プロパティのボタンの表示順を逆にし、真（true）の値を表すボタンが最初に表示されるようにします。デフォルトは false で、偽（false）を表すボタンが最初に表示されます。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

最初の例は、最初に表示されるラジオ ボタン グループに使用されるコードを示します。

```
<RadioButtonGroupControl property="value_order" labelWidth="2">
   <Layout gridWidth="4"/>
</RadioButtonGroupControl>
```

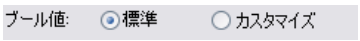
Layout 要素については、「詳細なカスタム レイアウト 」（ p. 207 ）に記載されています。

注 : ボタンおよびラベルの数は関連ノードの Properties セクションで定義されます。この場合、value_order プロパティの宣言内で列挙リストとして定義されます。宣言には、グループのラベルも含まれます。

```
<Property name="value_order" valueType="enum" label="カテゴリ入力の値順
  " labelKey="value_order.LABEL">
    <Enumeration>
        <Enum value="Ascending" label="昇順" labelKey="value_order.Ascending.LABEL"/>
        <Enum value="Descending" label="降順" labelKey="value_order.Descending.LABEL"/>
        <Enum value="DataOrder" label="データ順に使用" labelKey="value_order.UseDataOrder.LABEL"/>
    </Enumeration>
</Property>
```

2 番目の例では、標準またはカスタム設定を有効化するかどうかを制御するなど、以下のようにブール プロパティを制御するラジオ ボタン グループの falseLabel および trueLabel の使用について示します。

図 6-32
ブール プロパティを制御するラジオ ボタン グループ



実行するコードは次のとおりです。

```
<RadioButtonGroupControl property="boolean5" label="ブール値 5" labelKey="boolean5.LABEL"
    falseLabel="標準" falseLabelKey="boolean5.false.LABEL" trueLabel="カスタム"
    trueLabelKey="boolean5.true.LABEL" />
```

この場合、ボタンのラベルおよびグループのラベルを RadioButtonGroupControl 要素自体で定義されます。グループが関連するプロパティは、次のようにノードの Properties セクションで定義されます。

```
<Property name="boolean5" valueType="boolean" defaultValue="false"/>
```

サーバー ディレクトリ設定コントロール

ユーザーがサーバーのディレクトリを選択できる単一行のテキスト フィールドおよび関連ボタンを定義します。ディレクトリは、常に存在する必要があります。ユーザーは、モード設定に応じてファイルをディレクトリから開くか、ファイルをディレクトリに保存することができます。

図 6-33
サーバー ディレクトリ設定コントロール

ユーザーは、テキスト フィールドにディレクトリ パスまたはディレクトリ名を直接入力するか、隣接するボタンをクリックしてディレクトリを選択できるダイアログを表示することができます。

書式

```
<ServerDirectoryChooserControl controller_attributes mode="chooser_mode" >
   -- advanced custom layout options --
</ServerDirectoryChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

mode は、ユーザーがディレクトリを選択する、ダイアログに表示されるボタンを定義します。ボタンは次のいずれかです。

- open（デフォルト）は、［開く］ボタンを表示します。
- save は、［保存］ボタンを表示します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

```
<ServerDirectoryChooserControl property="directory1" label="サーバー ディレクトリ"
   labelKey="directory1.LABEL"/>
```

サーバー ファイル設定コントロール

ユーザーがサーバーのファイルを選択できる単一行のテキスト フィールドおよび関連ボタンを定義します。ディレクトリは、すでに存在する必要があります。ユーザーはノード設定によって、ファイルを開くか保存することができます。

図 6-34
サーバー ファイル設定コントロール

ユーザーは、テキスト フィールドにファイル パスまたはディレクトリ名を直接入力するか、隣接するボタンをクリックしてファイルを選択できるダイアログを表示することができます。

書式

```
<ServerFileChooserControl controller_attributes mode="chooser_mode" >
   -- advanced custom layout options --
</ServerFileChooserControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

mode は、ユーザーがファイルを選択する、ダイアログに表示されるボタンを定義します。ボタンは次のいずれかです。

- open（デフォルト）は、［開く］ボタンを表示します。
- save は、［保存］ボタンを表示します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

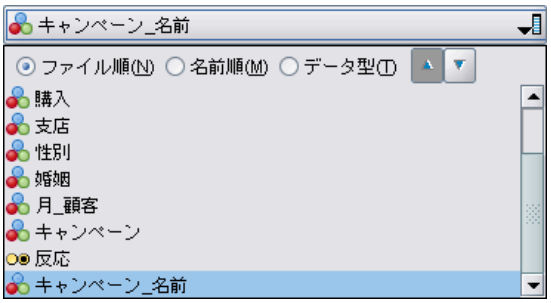
例

```
<ServerFileChooserControl property="file1" label="サーバー ファイル" labelKey="file1.LABEL"/>
```

単一フィールド設定コントロール

ユーザーがリストから単一フィールドを選択できるコントロールを定義します。

図 6-35
単一フィールド設定コントロール



ユーザーがこのコントロールをクリックすると、単一のフィールドを選択できるフィールドのリストが表示されます。

セットは、このノードで表示されるすべてのフィールドで構成されています。フィールドがこのノードの上流で詳細にフィルタリングされている場合、フィルタを通過したフィールドのみが表示されます。特定のストレージ タイプおよびデータ型のフィールドのみが選択できるよう指定することによって、リストをより詳細に制限することができます。

書式

```
<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>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

2 つの別の属性を指定してフィールドのリストをさらに制限することもできます。2 つの属性のいずれかは、次のリストから選択する必要があります。

- storage は、リスト内に表示されるフィールドのストレージ タイプを指定するリストのプロパティです。たとえば、storage="[integer real]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。可能なストレージ タイプのセットについては、データおよびストレージ タイプ p. 251 のテーブルを参照してください。
- onlyNumeric が true に設定されている場合、数値型ストレージのフィールドのみが一覧表示されます。
- onlySymbolic が true に設定されている場合、シンボル値のストレージ タイプのフィールド（文字列）のみが一覧表示されます。
- onlyDatetime が true に設定されている場合、日付と時間のストレージ タイプのみが一覧表示されます。

2 番目の属性を以下から選択する必要があります。

- types は、リスト内に表示されるフィールドのデータ型を指定するリストのプロパティです。たとえば、types="[range flag]" は、これらのストレージ タイプのフィールドのみ一覧表示されることを意味します。使用できるデータ 型のセットは次のとおりです。

  range

  flag

  set

  orderedSet

  numeric

  discrete

  typeless

- onlyRanges が true に設定されている場合、範囲のデータ型のフィールドのみが一覧表示されます。
- onlyDiscrete が true に設定されている場合、不連続な（フラグ、セット、データ型不明）のデータ型のみが一覧表示されます。

たとえば、storage="[integer]" および types="[flag]" を指定するコントロールでは、フラグ型である整数フィールドのみがリスト内に表示されます。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

注 : このコントロールは、ノードの定義にのみ使用されます。出力データモデルの定義内で複数フィールド設定を指定するには、次の形式を使用します。

```
<OutputDataModel mode="mode">
...
   <ForEach var="field" from="1" to="${integer}">
      <AddField name="${string}_${field}" fieldRef="${field_ref}" />
   </ForEach>
...
</OutputDataModel>
```

詳細は、 4 章 p. 81 出力データ モデル を参照してください。ForEach 要素については、「ForEach 要素による反復」（ p. 92 ）に記載されています。AddField については、「フィールドの追加」（ p. 89 ）に記載されています。

例

次の例では、前に示した単一フィールド設定コントロールを指定するために使用するコードを示します。

```
<SingleFieldChooserControl property="target"  storage="string" onlyDiscrete="true"/>
```

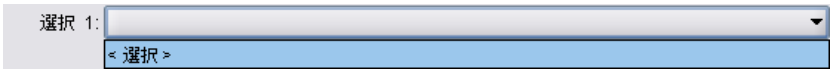
注 : 実際のリストの内容は、関連ノードの Properties セクションで定義されます。この場合、target プロパティの宣言内で定義されます。

```
<Property name="target" valueType="string" label="対象フィールド" labelKey="target.LABEL"/>
```

単一項目設定コントロール

ユーザーが値のリストから単一項目を選択できるコントロールを定義します。プロパティを、値のリストを持つカタログと関連付けます。 詳細は、 4 章 p. 56 カタログ を参照してください。

図 6-36
単一項目設定コントロール



書式

```
<SingleItemChooserControl controller_attributes catalog="catalog_name" >
   -- advanced custom layout options --
</MultiItemChooserControl
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

catalog（必須）は、関連するカタログの名前です。カタログが取得されるライブラリは、Execution セクションの Module 要素で指定されているライブラリです。 詳細は、 4 章 p. 78 モジュール を参照してください。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

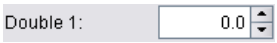
```
<SingleItemChooserControl property="selection1" catalog="cat1" />
```

このコントロールの使用に追加の説明および例は、「カタログ」 （ p. 56 ）を参照してください。

スピン コントロール

スピン コントロール（値を変更するための、上下の矢印の付いた数値型フィールド）を定義します。

図 6-37
スピン



書式

```
<SpinnerControl controller_attributes columns="integer" stepSize="increment"
   minDecimalDigits="number" maxDecimalDigits="number" >
   -- advanced custom layout options --
</SpinnerControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

columns は正の整数で、コントロールが占める文字の列数を指定します。デフォルトは 5 です。

stepSize は少数で、ユーザーが上下矢印のいずれかをクリックした場合にフィールド値が変更する数量を指定します。デフォルトは 1.0 です。

minDecimalDigits は、フィールド値に表示される小数部の桁数の最小値です。デフォルトは 1 です。

maxDecimalDigits は、フィールド値に表示される小数部の桁数の最大値です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

次の例では、前に示したスピン コントロールを指定するために使用するコードを示します。

```
<SpinnerControl property="double1" label="倍精度 1" labelKey="double1.LABEL"/>
```

数値型フィールドの内容、精度および有効な範囲は関連ノードの Properties セクションに定義されます。この場合、double1 プロパティの宣言に定義されます。

```
<Property name="double1" valueType="double" min="0" max="100"/>
```

テーブル コントロール

ノードのダイアログまたは出力ウィンドウに表示されるテーブルのレイアウト項目を定義します。

図 6-38
テーブル コントロール

書式

```
<TableControl controller_attributes rows="integer" columns="integer" columnWidths="list" >
   -- advanced custom layout options --
</TableControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

rows は正の整数で、画面上に表示されるテーブルの行数を指定します。デフォルトは 8 です。

columns は正の整数で、テーブルが占める文字の列数を指定します。デフォルトは 20 です。

columnwidths は、関連する列の幅を指定する値のリストを表示します。たとえば [30 5 10] の値は、列 1 が列 3 の 3 倍の幅になるよう指定します。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

前に示したテーブル コントロールを指定するコードは次のようになります。

```
<TableControl property="structure1" allowReorder="true" label="構造化"
 labelKey="structure1.LABEL" columnWidths="[20 6 10 10]">
   <ColumnControl column="0" editor="fieldValue" fieldControl="field1"/>
</TableControl>
```

テーブル コントロールの構造は、次のように特性ファイルの Common Objects セクションのプロパティの種類として定義されます。

```
<PropertyType id="shared_structure1" valueType="structure" isList="true">
   <Structure>
      <Attribute name="id" valueType="string" label="名前" labelKey="structure1.id.LABEL"/>
      <Attribute name="yesno" valueType="boolean" label="はい/いいえ" labelKey="structure1.yesno.LABEL"
       defaultValue="true"/>
      <Attribute name="count" valueType="integer" label="度数" labelKey="structure1.count.LABEL"
       defaultValue="0"/>
      <Attribute name="limit" valueType="double" label="制限" labelKey="structure1.limit.LABEL"
       defaultValue="0.0"/>
   </Structure>
</PropertyType>
```

ノード設定では、このプロパティの種類の識別子はプロパティの宣言によってテーブル コントロールの識別子と関連付けられます。

<Property name="**structure1**" type="**shared_structure1**"/>

スクリプトのノードを参照する場合、リストに大カッコ [] および構造に中カッコ {} を使用して、値をプロパティに設定することができます。たとえば、次のように structure1 プロパティの 2 つの構造のグリッドを設定できます。

set :node_ID.structure1 = [{"hello" true 4 0.21} {"bye" false 5 0.95}]

値の順番は、Attribute 定義が作成される順番と一貫している必要があります。

テキスト領域コントロール

複数行のテキスト エントリ フィールドを定義します。

図 6-39
テキスト領域コントロール

書式

<TextAreaControl controller_attributes rows="integer" columns="integer" wrapLines="true_false" >
  -- advanced custom layout options --
</TextAreaControl>

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

rows は正の整数で、テキスト領域が占める画面上の行数を指定します。デフォルトは 8 です。

columns は正の整数で、テキスト領域が占める文字の列数を指定します。デフォルトは 20 です。

wrapLines は、長いテキスト行に折り返しを使用する（true）または長いテキスト行を読む場合に水平スクロールを使用する（false）かどうかを指定します。デフォルトは true です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

最初に表示される例を作成するコードは、次のとおりです。

```
<TextAreaControl property="string2" label="文字列 2" labelKey="string2.LABEL"/>
```

この場合、入力データ型は関連ノードの Properties で定義されますが、テキスト領域のラベルはテキスト領域コントロールの宣言（string2 プロパティの宣言）で定義されています。

```
<Property name="string2" valueType="string" />
```

テキスト ボックス コントロール

単一行のテキスト エントリ フィールドを定義します。

図 6-40
テキスト ボックス コントロール

書式

```
<TextBoxControl controller_attributes columns="integer" >
   -- advanced custom layout options --
</TextBoxControl>
```

ここでの意味は次の通りです。

controller_attributes は、コントローラの属性 p. 177 で説明されているとおりです。

columns は正の整数で、テキストボックスが占める文字の列数を指定します。デフォルトは 20 です。

詳細なユーザー設定のレイアウト オプションでは、画面コンポーネントの位置および表示について、正確なコントロールを提供します。 詳細は、p. 207 詳細なカスタム レイアウト を参照してください。

例

最初に表示されるテキスト ボックスを作成するコードは、次のとおりです。

```
<TextBoxControl property="string1" label="文字列 1" labelKey="string1.LABEL"/>
```

テキスト ボックスのラベルおよび入力データ型は、関連ノードの Properties セクションに定義されます。この場合、string1 プロパティの宣言に定義されます。

```
<Property name="string1" valueType="string" />
```

プロパティ コントロールのレイアウト

この項では、ダイアログおよびウィンドウに使用される標準的なレイアウト方法およびこのレイアウトを変更して独自のカスタム レイアウトの取得方法について説明します。

標準的なコントロールのレイアウト

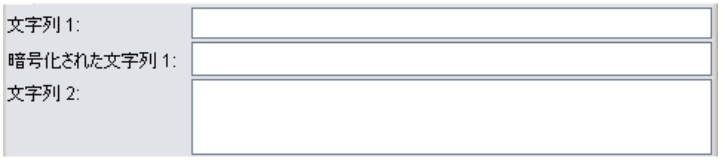
プロパティ パネルは、セルの 2 次元グリッドとしてみなすことができます。それぞれの行は高さが異なり、それぞれの列は幅が異なります。UI コンポーネントを複数の連続セルに割り当てることができますが、通常 UI コンポーネントは 1 つのセルにのみ割り当てられます。

デフォルトでは、1 つのプロパティ コントロールは 1 つの行に割り当てられ、各コントロールに 2 列ずつ割り当てられます。一方の列はラベル用、もう一方はコントロール コンポーネントまたはコンポーネント用です。ラベルを含む列は、最も広いラベルの幅にまで広がります。たとえば、特性ファイルに次の要素が指定されている場合

```
<TextBoxControl property="string1" label="文字列 1"/>
<PasswordBoxControl property="encryptedString1" label="暗号化文字列 1"/>
<TextAreaControl property="string2" label="文字列 2"/>
```

パネルは次のようになります。

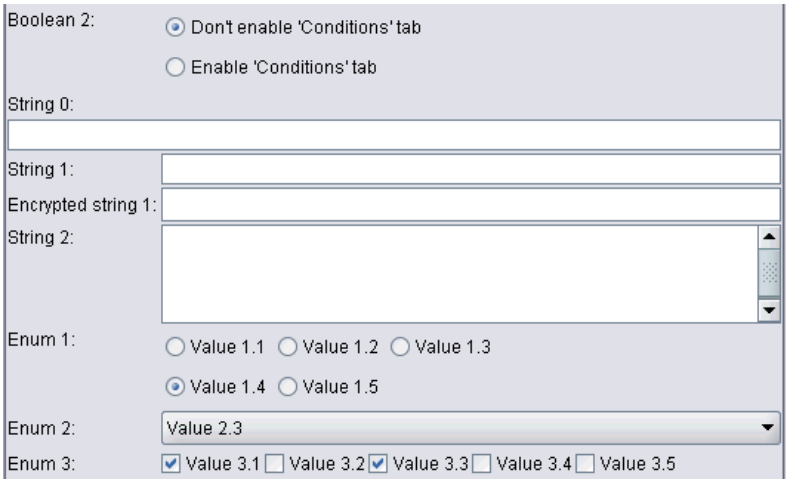
図 6-41
単純なプロパティ パネル



「:」文字は、ラベルの最後に自動的に追加されます。

複数のユーザー インターフェース コンポーネントを含むプロパティ コントロールは、これらのコンポーネントをレイアウトする表示されない四角形の領域を作成します。RadioButtonGroupControl と CheckBoxGroupControl 要素がこのようなコントロールの例となります。次の画面表示では、Boolean2、Enum1 および Enum3 というラベルのコントロールがこれを示しています。

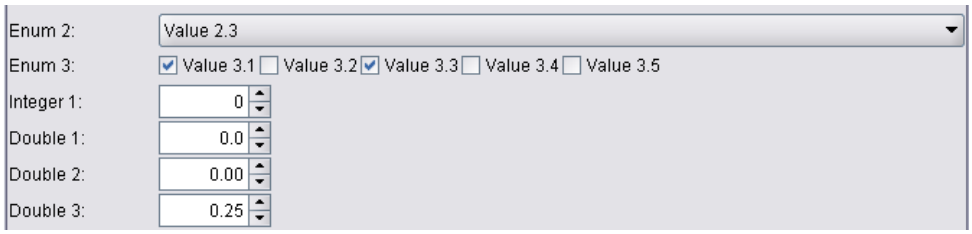
図 6-42
複数コンポーネント コントロールを持つプロパティ パネル



コンポーネントがレイアウトされる四角形の領域の形は、プロパティ コントロールによって異なります。さまざまなコントロールのレイアウトは、常に厳密に調整されているわけではありません。前図の Enum 1 と Enum 3 を比較してください。

一部のプロパティ コントロールには、コンポーネントの列を完全に満たすコンポーネントが含まれ、ウィンドウの幅が増減すると幅のサイズが変更されます。前の図で String 1、Encrypted string 1 および String 2 というラベルのコントロールで示されているとおり、この例では、TextBoxControl、PasswordBoxControl および TextAreaControl 要素でコントロールが指定されています。ただし、すべてのコンポーネントで実行されるわけではありません。たとえば、ウィンドウの幅が拡大されている場合でも、チェック ボックスおよびスピン コントロールは水平方向の領域の固定値のみをとります。

図 6-43
チェック ボックスおよびスピン コントロールを持つプロパティ パネル



コントロールのカスタム レイアウト

コントロールの標準レイアウトは、単純な方法または複雑な方法によって変更することができます。

単純なカスタム レイアウト

コントロール レイアウトをカスタマイズする単純な方法は、つぎの 3 つです。

- コンポーネントの上にラベルを配置する
- コントロールが配置される行数を変更する
- コントロールが配置される順序を変更する

コンポーネントの上へのラベルの配置

コントロールの labelAbove 属性を true に設定してコンポーネントの上の各行にラベルを配置できます。例 ：

```
<TextBoxControl property="string0" label="文字列 0"labelAbove="true"/>
<TextBoxControl property="string1" label="文字列 1"/>
<PasswordBoxControl property="encryptedString1" label="暗号化文字列 1"/>
```

コンポーネントの上にラベルを配置するとともに、実際の UI コンポーネントまたはコンポーネントは、画面のラベルの列に割り当てられます。これにより、対応するフィールドの上にラベル［文字列 0］が表示された次のパネルが作成されます。

図 6-44
各行にフィールド ラベルを持つパネル

行数の変更

デフォルトではラジオ ボタン グループとチェック ボックス グループは 1 行にレイアウトされ、ダイアログの幅はそれに合わせて調整されます。ラジオ ボタン グループまたはチェック ボックス グループには多数のオプションがあるため、ダイアログの幅が非常に広くなります。コントロールのレイアウトに使用される行数を変更して、これを回避できます。コントロール定義の rows 属性を該当する値に設定して行数を変更します。例 ：

```
<RadioButtonGroupControl property="enum1" label="列挙 1"rows="2"/>
```

これにより、2 行にラジオ ボタン グループが配置されたパネルが作成されます。

図 6-45
2 行にラジオ ボタン グループが配置されたパネル

コントロールの順序の変更

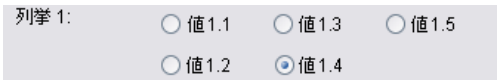
ラジオ ボタンおよびチェック ボックス グループの場合、enum 値のコントロールがパネルに追加される順序を変更することもできます。

デフォルトでは、前述の例のようにコントロールは行順に追加され、1 番目、2 番目、3 番目の値が最初の行に、4 番目および 5 番目の値が 2 番目の行に追加されます。代わりに、layoutByRow を false に設定して、指定された行数内でコントロールを列順に追加することができます。例 :

```
<RadioButtonGroupControl property="enum1" label="列挙 1" rows="2"layoutByRow="false"/>
```

値は 2 行にわたって表示されますが、1 番目および 2 番目の値が最初の列に、3 番目および 4 番目の値が 2 番目の列に、そして 5 番目の値は 3 番目の列に追加されます。

図 6-46
列順にラジオ ボタン グループが配置されたパネル



2 つのラジオ ボタンとして表示されるブール プロパティの場合、デフォルトの順序によって、「真 (True)」のボタンの前に「偽 (False)」のボタンが表示されます。trueFirst 属性を true に設定して、この順序を逆にすることができます。

useSubPanel 属性を false に設定して、ラジオ ボタンおよびチェック ボックス グループでサブパネルを使用できないようにすることもできます。ただし、Layout 要素と組み合わせて使用されない限り、望まないレイアウトが行われる場合があります (「Layout 要素を使用したコントロール位置の正確な指定」 (p. 208) を参照。

詳細なカスタム レイアウト

各コントロールの宣言内で、さまざまな要素を使用して複雑なコントロールのレイアウトを指定することができます。次のことが可能です。

- Layout 要素を使用して、画面上のコントロールを正確に指定する

- Enabled 要素を使用して、表示の特性を制御する
- Visible 要素を使用して、画面コンポーネントが表示されるかどうかを制御する

Layout 要素を使用したコントロール位置の正確な指定

次のように、明示的な Layout 要素を指定してコントロールと関連付けることによって、正確なレイアウトの位置を指定することができます。

書式

```
<property_control ... >
   <Layoutattributes
    --- cell specification ---
       ...
</property_control>
```

ここでの意味は次の通りです。

property_control プロパティ コントロールの 1 つです（「プロパティ コントロール設定」（ p. 167 ）を参照）。

attributes は、以下の属性のいずれかです。

テーブル 6-6
レイアウト属性

属性	値	説明
anchor	north northeast east southeast south southwest west northwest center	コントロールのアンカー ポイントを定義します。
columnWeight	0.0-1.0	ウィンドウの水平方向のサイズ変更がコントロールの幅に与える影響を定義します。パネル内のすべての columnWeight 属性の合計は、1.0 を超えることはできません。
fill	none horizontal vertical both	コントロールが割り当てられたセルに入力するかどうか、どのように入力するかを定義します。
gridColumn	整数または 0	コントロールのレイアウトが開始する最初の列を定義します。

属性	値	説明
gridHeight	integer	コントロールが配置される行数を定義します。0 の値（デフォルト）を使用すると、残りの行すべてにコントロールを割り当てます。
gridRow	整数または 0	コントロールのレイアウトが開始する最初の行を定義します。デフォルトでは、グリッドの行のインデックスが自動的に増加します。
gridWidth	integer	コントロールが配置される列数を定義します。0 の値（デフォルト）を使用すると、残りの列すべてにコントロールを割り当てます。
leftIndent	integer	デフォルトの位置からコントロールをインデントするピクセル数を定義します。
rowWeight	0.0-1.0	ウィンドウの垂直方向のサイズ変更がコントロールの高さに与える影響を定義します。パネル内のすべての rowWeight 属性の合計は、1.0 を超えることはできません。

cell specification を使用すると、画面上のコントロールの正確な位置を指定します。形式は次のとおりです。

```
<Cell row="integer" column="integer" width="integer" />
```

ここでの意味は次の通りです。

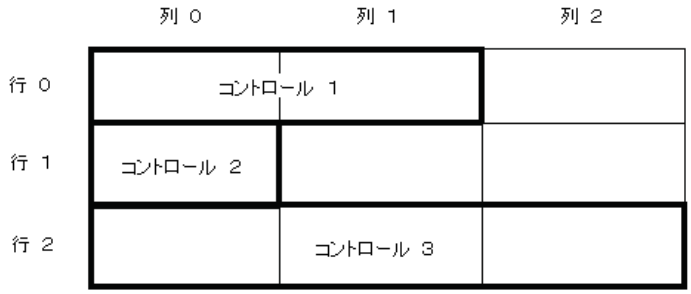
row（必須）は負ではない整数で、コントロールが開始する行の位置を指定します。

column（必須）は負ではない整数で、コントロールが開始する列の位置を指定します。

width（必須）は負ではない整数で、コントロールが占める画面上ののグリッド列の数を指定します。

たとえば、画面のグリッドが 3 列および 3 行と仮定すると、コントロールのカスタム レイアウトの形式は次のようになります。

図 6-47
セルを使用したコントロールのレイアウトの例



次のセル設定の Layout 要素が必要です。

```
<Layout ...>
   <Cell row="0" column="0" width="2">
   <Cell row="1" column="0" width="1">
   <Cell row="2" column="0" width="3">
</Layout>
```

次の例では、Layout 要素の使用方法について詳細に説明しています。

例 :テキスト フィールドを有効化するチェック ボックス

この例では、画面の同じ行にあるテキスト フィールドを有効化するチェック ボックスの使用について説明します。

チェック ボックスを使用して同じ行の別のコントロールを有効化する場合、単純な Layout を使用してコントロールを正しく表示する必要があります(注 : コントロールを有効化および無効化するメカニズムは 「Enabled 要素を使用した表示の特性の制御」 (p. 217) に説明されています)。

画面で次を実行すると仮定します。

図 6-48
テキスト フィールドを有効化するチェック ボックス

以下の 2 つのコントロールがあります。

- テキスト フィールドのラベルとしても動作するラベルを持つチェック ボックス
- テキスト フィールド

開始ポイントは、2 つのコントロールの通常の宣言です。

```
<CheckBoxControl property="boolean3" label="チェック ボックス 3"/>
<TextBoxControl property="string3" label="文字列 3"/>
```

パネルは次のようになります。

図 6-49
各行のチェック ボックスおよびテキスト フィールド

まず、テキスト フィールド ラベル [文字列 3] を非表示にするとします。テキスト フィールド コントロールの showLabel 属性を false に設定して非表示にします。

```
<CheckBoxControl property="boolean3" label="チェック ボックス 3"/>
<TextBoxControl property="string3" label="文字列 3"showLabel="false"/>
```

テキスト フィールドは、ラベルが以前占めていた領域を満たすまで拡大します。

図 6-50
ラベルのないチェック ボックスおよびテキスト フィールド

テキスト フィールドをチェック ボックスと同じ行に表示するとします。このためには、CheckBoxControl 要素内に Layout 要素を追加して、行の増分を 0 に設定します（デフォルトでは、行はコントロールごとに 1 ずつ増加します。）。

```
<CheckBoxControl property="boolean3" label="チェック ボックス 3">
   <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false"/>
```

ただし、表示は次のようになります。

図 6-51
チェック ボックスに重複するテキスト フィールド

テキスト フィールドは 1 行上に移動しますが、行全体に占められているため、チェック ボックスに重複します。

注 ： 次のように表示された場合、

図 6-52
テキスト フィールドに重複するチェック ボックス

チェック ボックスはテキスト フィールドの後に描画されるため、テキスト フィールドの最初の数文字は表示されません。

どちらのオブジェクトを最初に描画した場合でも、複数の UI コンポーネントを同じセルに割り当てると望まない動作または未定義の動作が生じるため、避ける必要があります。この問題を解決するには、2 番目の Layout 要素を追加する必要があり、次の例では TextBoxControl 要素内でテキストフィールドに画面の 2 番目の列から開始させます。

```
<CheckBoxControl property="boolean3" label="チェック ボックス 3">
   <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false">
   <Layout gridColumn="1"/>
</TextBoxControl>
```

ただし、結果が次のようになるため、これは部分的な解決方法です。

図 6–53
短いテキスト フィールドの正しい配置

2 つのコントロールは正しく配置されますが、テキスト フィールドは短くなります。問題は、カスタム レイアウトがコントロールと関連すると、各種のコントロールに関連する「スマートな」デフォルトを上書きすることです。この場合、Layout 要素のセルを満たすデフォルトの操作（コンポーネントが使用できるセルをどのように満たすか）は、使用可能なセルを満たすことではなく、画面上で可能な限り小さい領域になります。これを変更するには、テキスト フィールドを水平方向の領域を満たすよう指示します。

```
<CheckBoxControl property="boolean3" label="チェック ボックス 3">
   <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false">
   <Layout gridColumn="1"fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

Java を割り当てて領域を正しく満たすよう、小さい columnWeight 値を追加する必要があります。

これにより、希望のレイアウトが指定されます。

図 6–54
テキスト フィールドを有効化するチェック ボックス

正しく表示されているように思われますが、処理する必要のある問題が 1 つあります。同じ行に別のコントロールを配置する場合でも、チェック ボックスを正しく行全体に配置します。問題は、チェック ボックスのラベルが比較的短いために正しく表示されず、重複しないようパネルの他のラベル（図では表示されていません）によって 2 番目の表示列を

移動させていることです。チェック ボックスの食べるをより長くすると、問題が明確になります。

```
<CheckBoxControl property="boolean3" label="使用していたラベルより長い
ラベルを持つチェック ボックス 3">
   <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

これにより、以下のようになります。

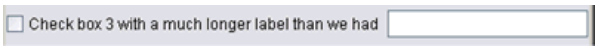
図 6–55
テキスト フィールドが重複する長いチェック ボックスのラベル

チェック ボックスが 1 列あたりの可能な幅を制限するよう指示する必要があります。

```
<CheckBoxControl property="boolean3" label="使用していたものより長いラベルを持つチェック ボックス 3">
   <Layout rowIncrement="0"gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

これにより、希望したとおりになります。

図 6–56
完全に表示された長いチェック ボックスのラベル



例 :ラジオ ボタン グループおよびテキスト フィールド

この例では、各ラジオ ボタン グループをそれぞれのテキスト フィールドと関連付ける方法について説明します。

以下のようなパネルを定義する必要があるとします。

図 6–57
テキスト フィールドを含むラジオ ボタン グループ

この例では、次の 4 つのコントロールがあります。

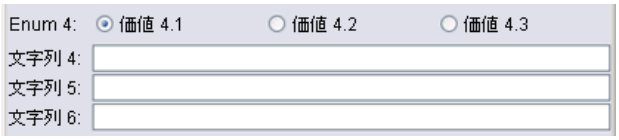
- 3 つの値の列挙リストのラジオ ボタン グループ
- 各値に 1 つずつ、3 つのテキスト フィールド

前述の例のように、コントロールの単純な宣言で開始されます。

```
<RadioButtonGroupControl property="enum4" label="列挙 4"/>
<TextBoxControl property="string4" label="文字列 4"/>
<TextBoxControl property="string5" label="文字列 5"/>
<TextBoxControl property="string6" label="文字列 6"/>
```

これにより、次のように表示されます。

図 6-58
テキスト フィールドおよびラベルを含むラジオ ボタン グループ



ラジオ ボタンのラベルを使用してテキスト フィールドを識別するには、まず 3 行で 1 列にラジオ ボタンを配置し、テキスト フィールドのラベルを非表示にします。

```
<RadioButtonGroupControl property="enum4" label="列挙 4"rows="3"/>
<TextBoxControl property="string4" label="文字列 4"showLabel="false"/>
<TextBoxControl property="string5" label="文字列 5"showLabel="false"/>
<TextBoxControl property="string6" label="文字列 6"showLabel="false"/>
```

これにより、次のように表示されます。

図 6-59
1 列に並ぶラジオ ボタンおよびテキスト フィールド

ここで 1 つの問題があります。ラジオ ボタン グループのラベルが最初のラジオ ボタンに合わせて整列していません。この問題は後で解決します。テキスト フィールドを対応するそれぞれのラジオ ボタンの行に大まかに合わせます。

必要な手順は、例 1 と同様、以下のとおりです。

- ラジオ ボタン グループの行の増分を 0 に変更します。

- グリッド幅を制限して、テキスト フィールドとラジオ ボタンが重複しないようにします。
- ラジオ ボタンと同じ行の各テキスト フィールドを配置します。

前述の例と同様、Layout 要素を追加します。この場合、次のように特性ファイルを変更します。

```
<RadioButtonGroupControl property="enum4" label="列挙 4" rows="3">
   <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="文字列 4" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="文字列 5" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="文字列 6" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

この時点で、次のように表示されます。

図 6-60
ラジオ ボタンに重複するテキスト フィールド

例 1 で使用したものとまったく同じ Layout 要素を使用しましたが、何が起こったのでしょう ?

それは、例 1 のチェック ボックス コントロールとは異なり、ラジオ ボタン グループ（多くのコントロールと同様）には実際のコントロールとともに個別のラベルがあるためです。つまりラジオ ボタン グループには追加の列が必要であるため、テキスト フィールドが後の列、列 1 ではなく列 2 から開始するよう指示する必要があります。テキスト フィールドの Layout 要素で、gridColumn 値を 2 に設定します。

```
<RadioButtonGroupControl property="enum4" label="列挙 4" rows="3">
   <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="文字列 4" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="文字列 5" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="文字列 6" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
```

```
</TextBoxControl>
```

テキスト フィールドのグリッド列は 2 に増加しますが、ラジオ ボタン グループのグリッド幅は 1 から増加しません。これは、プロパティ コントロールで、ほとんどの Layout 属性がコントロールのラベルではなく、コントロールの編集可能な部分を構成する UI コンポーネントにのみ影響を与えるためです。

次のように表示されます。

図 6-61
ラジオ ボタンに重複していないテキスト フィールド

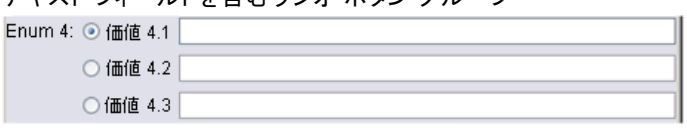
希望のレイアウトにかなり近づきました。ただし、ラジオ ボタンおよびテキスト フィールド間の位置合わせの問題がいくつか残っています。

主な問題は、ラジオ ボタンが各サブパネルに配置されているため、ラジオ ボタンとテキスト フィールドの間に実際のレイアウトの関係がないことです。サブパネルを使用して、次のようにラジオ ボタン グループを停止する必要があります。

```
<RadioButtonGroupControl property="enum4" label="列挙 4" rows="3"useSubPanel="false">
   <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="文字列 4" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="文字列 5" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="文字列 6" showLabel="false">
   <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

これにより、希望のレイアウトが設定されました。

図 6-62
テキスト フィールドを含むラジオ ボタン グループ



Enabled 要素を使用した表示の特性の制御

Enabled 要素を使用して、通常は特定の条件を満たすかどうかによって、コントロールを有効化または無効化することができます。

パネルおよびプロパティ コントロールには、さまざまな表示特性を決定するために関連付けられた条件がある場合があります。たとえば、チェック ボックスを使用して関連するテキスト フィールド有効化することができます。またはラジオ ボタンによって非表示フィールドのグループを表示されるようにすることもできます。

ユーザー インターフェイスの条件は通常、プロパティではなく別のコントロールの値に基づきます。プロパティに基づく条件は、変更が基本のオブジェクト（ノード、モデル出力、またはドキュメント 出力）に適用された場合にのみ有効です。ユーザー インターフェイスでは、関連するコントロールが変更された後すぐにコントロールを有効化する必要があります。

書式

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

Condition 要素で、条件を指定してコントロールを有効化するかどうかを決定するために検定します。

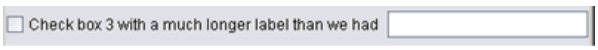
And、Or、および Not 要素を使用して、複合条件を指定できます。

詳細は、 4 章 p.98 Conditions を参照してください。

例 :単純な条件によるコントロールの有効化

例 :テキスト フィールドを有効化するチェック ボックス p.210 で、ボックスが選択された場合にテキスト フィールドを有効化できるチェック ボックスを作成します。

図 6-63
完全に表示された長いチェック ボックスのラベル



チェック ボックスが選択されるとすぐにテキスト フィールドを有効化し、基本オブジェクトのプロパティが変更されるとテキスト フィールドを無効化する必要があるとします。これを実現するには、次のように Enabled 条件を追加する必要があります。

```
<CheckBoxControl property="boolean3" label="使用していたものより長いラベルを持つチェック ボックス3">
   <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="文字列 3" showLabel="false">
   <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
      <Enabled>
         <Condition control="boolean3" op="equals" value="true"/>
      </Enabled>
</TextBoxControl>
```

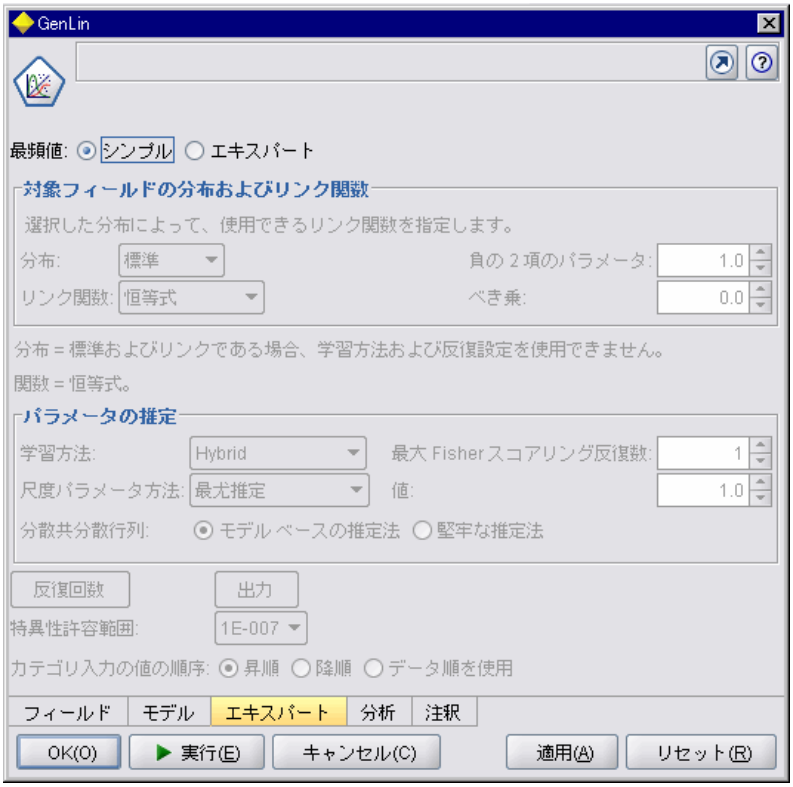
これにより、チェック ボックスに関連するブール値が真の場合にのみ、テキスト フィールドが有効になります。

例 :複雑な条件によるコントロールの有効化

複雑な条件のコーディングを理解するために、CLEF を使用して作成された一般化線型ノードのダイアログ タブの 1 つを参照します。

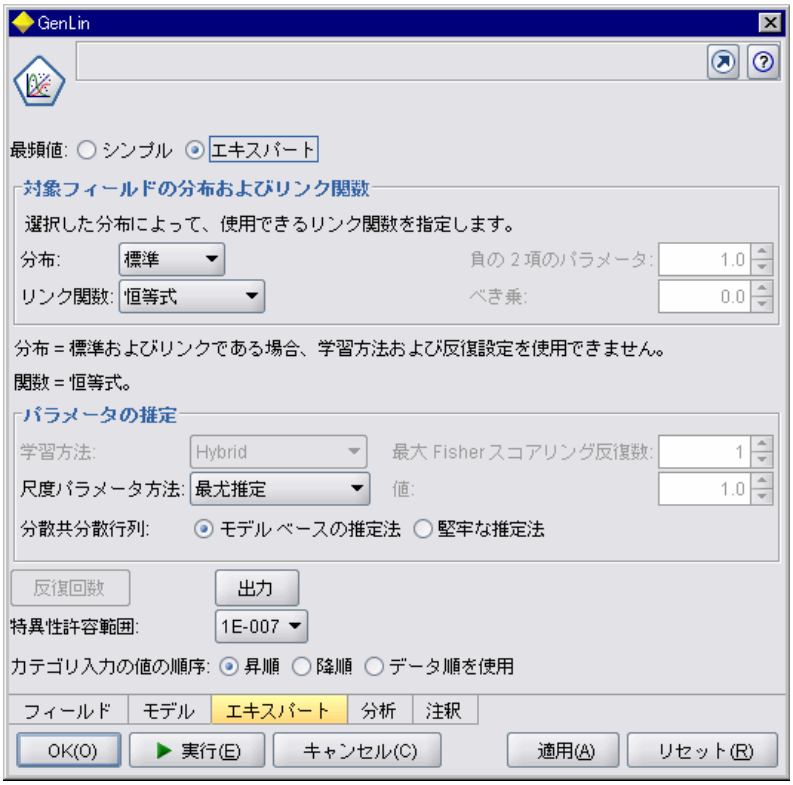
ノード ダイアログには［エキスパート］タブがあり、ユーザーにモデルの詳細な知識を提供するオプションが含まれています。タブのすべてのオプションは、最初は無効です。

図 6-64
すべてのオプションが無効な [エキスパート] タブ



[**モード**] チェック ボックスを [**エキスパート**] に設定すると、次のようにさまざまなオプションが有効になります。

図 6-65
[モード] を [エキスパート] に設定した [エキスパート] タブ

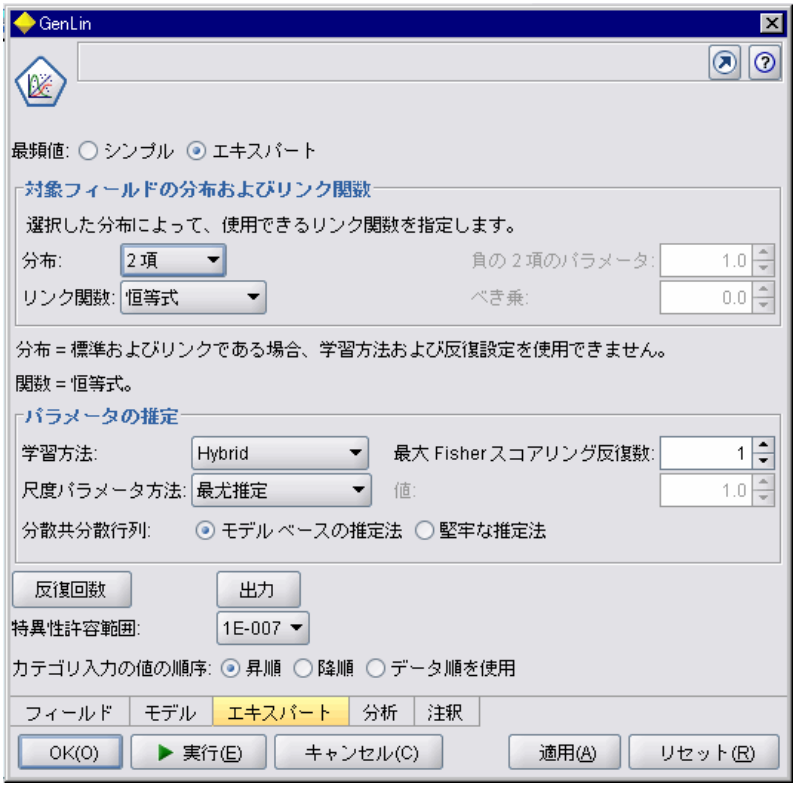


ただし、ダイアログの下部にある［**反復**］コントロールなど、無効なオプションもあります。このコントロールは、次の条件の**両方**が真の場合にのみ無効です。

- ［**分布**］が［**正規分布**］に設定されている
- ［**リンク関数**］が［**恒等式**］に設定されている

この組み合わせは実際エキスパート モードの［エキスパート］タブのデフォルト設定で、これらの選択情報のいずれかの設定を変更すると、［**反復**］を有効にします。

図 6-66
[反復] ボタンを有効にする分布設定への変更



上記を実行するコードは、次のように [反復] ボタンの PropertiesSubPanel 宣言に含まれています。

```
<PropertiesSubPanel buttonLabel="反復..." 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>
```

外部の And セクションの Condition 要素は、変更が行われる前に［**モード**］が［**エキスパート**］に設定されるよう指定します。この条件が真の場合、Not セクションは、内部の And セクションの「両方」の条件が満たされている場合にのみボタンが有効にならないように（つまり、無効になるように）指定します。エキスパート モードでは、［**分布**］または［**リンク関数**］のいずれかがデフォルト値以外に設定されている場合、［**反復**］が有効になります。

Visible 要素を使用した表示の特性の制御

条件を使用して、特性の環境に応じてコントロールを表示または非表示にすることができます。Visible 要素によって行われます。

書式

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

Condition 要素で、条件を指定してコントロールを表示するかどうかを決定するために検定します。

And、Or、および Not 要素を使用して、複合条件を指定できます。

詳細は、 4 章 p.98 Conditions を参照してください。

例

次の例では、source_language 条件が満たされている場合にのみ、指定されたプロパティ パネルを表示します。

```
<PropertiesPanel>
   <Visible>
      <Condition control="source_language" op="equals" value="eng" />
   </Visible>
     ...
</PropertiesPanel>
```

カスタム出力ウィンドウ

モデル出力、ドキュメント出力、インタラクティブな出力オブジェクト（ノード以外）の場合、拡張によってデフォルト出力ウィンドウをカスタムウィンドウを完全に置き換えることはできません。標準の java.awt.Frame クラスとして実装されていますが、.

カスタム ウィンドウを提供するには、次のように Java クラスを UserInterface 要素の frameClass 属性として指定します。

```
<DocumentOutput id="my_modelling_node" type="modelBuilder" ...>
   <Properties>
      <Property name="use_custom_type" valueType="boolean" .../>
       ...
   </Properties>
   <UserInterfaceframeClass="com.myextension.MyOutputFrame"/>
   ...
 </DocumentOutput>
```

指定されたクラスは、CLEF のクライアント側の API で定義された ExtensionObjectFrame インターフェイスを実装する必要があります。これは次のようなウィンドウのライフ サイクルをカバーします。

- 基本の java.awt.Frame へのアクセス
- 出力オブジェクトおよびセッションへのアクセスなど、ウィンドウの初期化
- オブジェクトを保存または削除しようとする場合のウィンドウと基本オブジェクトの同期化
- ウィンドウの削除

詳細は、 9 章 p.241 クライアント側 API クラス を参照してください。

章 7

ヘルプ システムの追加

ヘルプ システムの種類

CLEF 拡張子を作成する場合、通常オンラインの ヘルプ システムを使用して、拡張子の使用方法を説明する必要があります。CLEF は、次のタイプのヘルプ システムをサポートします。

- HTML Help
- JavaHelp

HTML Help

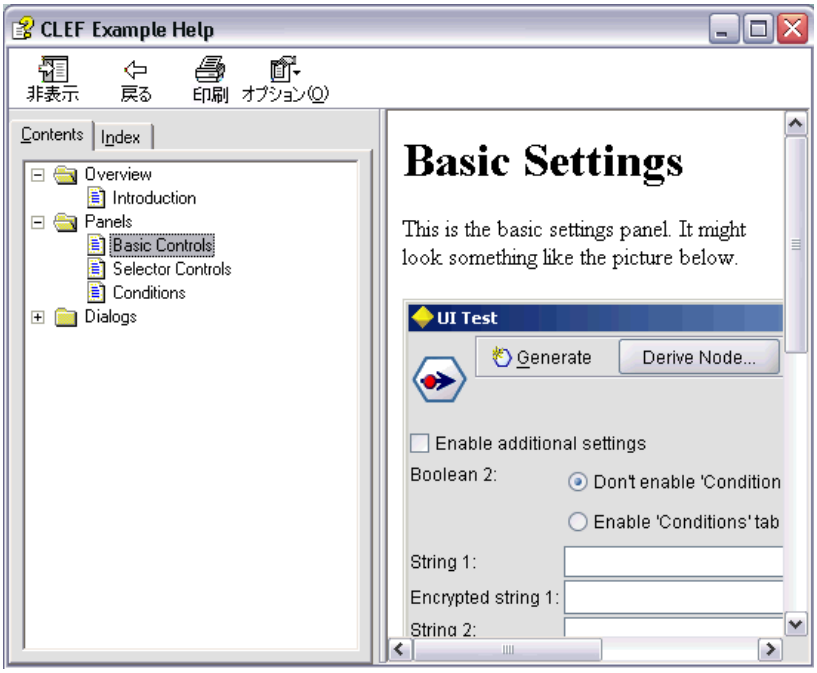
HTML Help は、Windows プラットフォーム上でのみ動作する、Microsoft が開発した独自の形式です。HTML Help システムは、.chm を持つ単一ファイルとして圧縮された形式にコンパイルされた .htm または .html ファイルで構成されています。IBM® SPSS® Modeler の独自のヘルプ システムは、HTML Help 形式で提供されます。

HTML Help ファイルは、目次、インデックス、全文検索機能をサポートします（用語集をポップアップ ウィンドウとして実装できます)。HTML エディタまたは商用のヘルプ オーサリング ツールを使用して、ソース .htm または .html トピック ファイルを作成できます。.chm ファイルを作成するには、Microsoft ダウンロード センター Web サイトから無料ダウンロード可能な HTML Help Workshop を使用します（.chm ファイル作成の詳細は、HTML Help Workshop ヘルプ システムを参照してください)。また、HTML Help 形式をサポートするヘルプ オーサリング ツールを使用して、.chm ファイルに使用されるトピック ファイルおよびグラフィック ファイルをコンパイルすることもできます。

HTML Help ブラウザは、次のようになります。

© Copyright IBM Corporation 1994, 2012.

図 7-1
HTML Help ブラウザ



JavaHelp

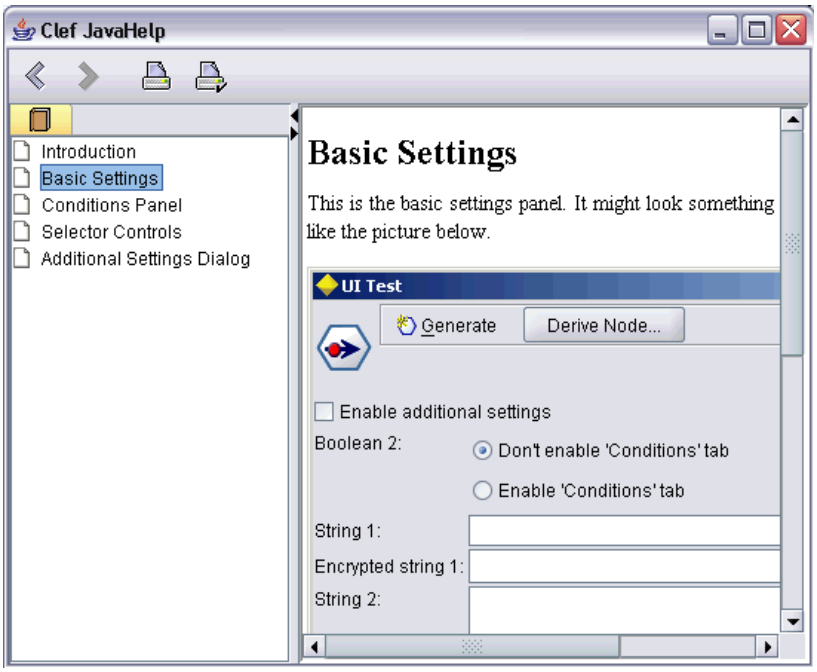
JavaHelp は、Java をサポートするすべてのプラットフォームで動作する、Sun Microsystems が開発したオープン ソースのヘルプ形式です。JavaHelp システムは、次のファイルで構成されています。

- ソース .htm または .html トピック ファイル
- トピックで使用されるすべてのグラフィック ファイル
- ヘルプ システムを制御するヘルプセット ファイル（拡張子 .hs）
- トピック ID をトピック ファイルを関連付け、ヘルプ トピックを表示するウィンドウを定義するために使用する map.xml ファイル
- インデックス エントリを含む index.xml ファイル
- 目次のエントリを含む toc.xml ファイル

JavaHelp は、目次、インデックス、全文検索、用語集機能をサポートします。HTML エディタまたは商用のヘルプ オーサリング ツールを使用して、ソース .htm または .html ファイルを作成できます。また、Sun Developer Network の Web サイトから無料ダウンロードできる JavaHelp ソフトウェアも必要です（詳細は、この Web サイトで入手可能な『JavaHelp System User's Guide』を参照してください）。

この JavaHelp ブラウザは、次のようになります。

図 7-2
JavaHelp ブラウザ



ヘルプ システムの実装

この項では、設定ファイルのヘルプ システムの関連するコンポーネントの定義方法について説明します。

ヘルプ システムの場所および種類の定義

拡張子に使用されるヘルプ システムの種類がある場合、拡張子に対する設定ファイルのリソース セクションの HelpInfo の要素で定義されます。

書式

```
<Resources>
 ...
   <HelpInfo id=" name" type=" help_type" path=" help_path" helpset=" helpset_loc" default=" default_topicID" />
 ...
</Resources>
```

ここでの意味は次の通りです。

id（必須）は、この拡張子に対するヘルプ情報の識別子です。

type（必須）は、ヘルプの種類を表し、次のいずれかを指定します。

- htmlhelp– path 属性に識別される .chm ファイルに含まれた HTML Help。
- javahelp– helpset 属性で識別されるヘルプセット（.hs）ファイルを、ヘルプ ソースおよび関連ファイルと合わせて使用する JavaHelp。

ヘルプの種類が htmlhelp である場合、次の追加の属性が必要です。

- path– ヘルプ システムを含む .chm ファイルの場所（設定ファイルに関連）と名前。

ヘルプの種類が javahelp である場合、次の追加の属性が必要です。

- helpset– 使用される .hs ヘルプセット ファイルの場所（設定ファイルに関連）と名前。
- default– 特定のタブにトピックが指定されていない場合に表示されるデフォルトのトピックの識別子。

HelpInfo の要素が指定されていない場合、この拡張子に関連するヘルプはありません。

例

最初の例では、HTML Help の HelpInfo の要素を説明します。

```
<HelpInfo id="help" type="htmlhelp" path="help/mynode.chm" />
```

JavaHelp システムでは次のようになります。

```
<HelpInfo id="help" type="javahelp" helpset="help/mynode.hs"/>
```

JavaHelp の場合、関連ファイル（画像、マップ ファイル、インデックス、コンテンツ ファイル）は、.hs ヘルプセット ファイルと同じフォルダに配置されている必要があります。

表示する特定のヘルプ トピックの指定

ユーザーがノードのダイアログ、特定のタブ、プロパティのサブパネルでヘルプを起動する場合に表示する特定のヘルプ トピックを指定できます。ノード、タブ、またはプロパティのサブパネルの指定の helpLink 属性によって実行されます。

helpLink 属性が指定されていない場合、ヘルプ システムのデフォルト トピックが、ユーザーがヘルプを起動する際に表示されます。

詳細は、「ノード」（ p. 66 ）, 「タブ」（ p. 155 ）および 「プロパティ サブパネル」（ p. 172 ）のhelpLink 属性の説明を参照してください。

例

この例では、HTML Help を使用していると仮定し、ユーザーがヘルプを選択する場合にフォーカスがあるウィンドウによってさまざまなコンテキスト トピックを表示できる方法を示します。

```
<Resources>
   ...
  <HelpInfo id="help" type="htmlhelp" path="help/mynode.chm"/>
   ...
</Resources>
...
<Node id="mynode" scriptName="my_node" type="dataTransformer" palette="recordOp"
 label="ソーター" description="データ ファイルをソート" >
 ...
  <Tabs defaultTab="1">
    <Tab label="基本的なコントロール" labelKey="basicControlsTab.LABEL"
     helpLink="basic_controls.htm">
       <PropertiesPanel>
        ...
          <PropertiesSubPanel buttonLabel="追加設定..."
           buttonLabelKey="AdditionalOptions.LABEL" dialogTitle="追加設定"
           dialogTitleKey="AdditionalOptionsDialog.LABEL"helpLink="addsettingsdlg.htm">
 ...
    </Tab>
    <Tab label="選択コントロール" labelKey="selectorControlsTab.LABEL"
     helpLink="selector_controls.htm">
 ...
    </Tab>
 ...
</Node>
```

この場合、[基本コントロール] タブにフォーカスがありユーザーがヘルプを選択すると、mynode.chm ヘルプ ファイルの basic_controls.htm のトピックが表示されます。その後ユーザーが [追加設定] ボタンをクリックして [追加設定] ダイアログを開き、そのダイアログの [ヘルプ] を選択すると、addsettingsdlg.htm のトピックが表示されます。ユーザーが [追加設定] ダイアログを閉じる場合、[選択コントロール] タブを選択して [ヘルプ] を再度選択すると、selector_controls.htm のトピックが表示されます。

JavaHelp の場合、helpLink 属性の値は、map.xml ファイルの target 属性の値と一致する必要があります。たとえば、map.xml ファイルに

```
<map version="1.0">
 ...
  <mapID target="basic_controls" url="basic_controls.htm"/>
 ...
</map>
```

が含まれる場合、対応する helpLink 属性に次の値を指定する必要があります。

helpLink="**basic_controls**"

これは、JavaHelp が呼び出される場合、target 属性の値を読み込み、関連する url 値にマップして表示する適切なファイルを検出するためです。

章 8

ローカライゼーションとアクセシビリティ

はじめに

ローカライゼーションは、特定のロケールのソフトウェア、ヘルプ、マニュアルを適合させるプロセスを参照しています。これには、ユーザーインターフェイス、ヘルプおよびマニュアルの翻訳および適切なロケールのシステムの検定が含まれます。自身の地域以外の地域に拡張子を配布する場合、その拡張子のローカライズされたバージョンを配布することができます。

ここでは、**アクセシビリティ**はユーザー インターフェイスの機能追加を参照し、視覚障害または手先の動作の制限など、特定の障害を持つユーザーにとってシステムへのアクセスが容易にできるようにします。

ローカライゼーション

IBM® SPSS® Modeler は、世界の多くの地域にローカライズされています。サポートされたいずれかの言語について、ユーザーは Windows の地域オプションを自身のロケールに設定すると、たとえば次のような標準 IBM® SPSS® Modeler UI コンポーネントが該当する言語で表示されます。

- システム メニューとメニュー エントリ
- システム ボタン（生成、OK、実行、キャンセル、適用、リセット）
- 標準的なダイアログ ボックスのタブ（使用される場合、注釈およびデバッグ）
- エラーおよびシステム メッセージ（「このオブジェクトは保存されていません。」など）

拡張子がこれらの標準 SPSS Modeler コンポーネントを使用している場合、選択した言語がサポートされていればその言語で自動的に表示されます。

拡張子の他のコンポーネントに対し、CLEF ではローカライズを支援する機能を提供しています。ローカライズできるのは次のとおりです。

- ノード名（パレットおよびキャンバス上）
- モデル名（マネージャ領域の［モデル］タブ）
- ドキュメント名（マネージャ領域の［出力］タブ）
- アクションに関連するアイコン イメージの場所

© Copyright IBM Corporation 1994, 2012.

- ツールヒント テキスト
- ヘルプ システム
- ノード ダイアログ
  - タイトル バー テキスト
  - カスタム メニューとメニュー エントリ
  - フィールド、プロパティ、ボタンおよびタブのラベル
  - 静的テキスト
- システムおよびエラー メッセージ

テキスト文字列は、項目を翻訳した場合により長いテキストを使用できるよう、短くする必要があります。

システム メッセージおよびエラー メッセージは、設定ファイル、プロパティ ファイルおよびサーバー側 API を組み合わせてローカライズできます。 詳細は、 9 章 p.264 状況詳細ドキュメント を参照してください。

プロパティ ファイル

ローカライズできるテキスト文字列は**プロパティ ファイル**と呼ばれるファイルにテキスト文字列として保存されます。ローカライゼーションのリソース バンドルを保存するために標準 Java 形式を使用します。それぞれのプロパティ ファイルは、一連のレコード、拡張子のローカライズされた各項目のレコードで構成されています。各レコードのフィールドは、設定ファイルの labelKey 属性に対応しており、CLEF はプロパティ ファイルから対応するテキスト文字列を読み込み、適切な場所に表示することができます。

プロパティ ファイルは拡張子 .properties を持つ必要があり、関連するノードの設定ファイルと同じディレクトリにある必要があります。IBM® SPSS® Modeler は最初に次の名前を持つデフォルトのプロパティ ファイルを検索します。

```
path.properties
```

この場合、path は、プロパティ バンドルを定義する（Resources セクション内の）Bundle 要素の path 属性の値です。次に例を示します。

```
<Bundle id="bundle" path="my_resources"/>
```

デフォルトのプロパティ ファイルがない場合、SPSS Modeler は設定ファイルの定義からテキスト文字列を読み込みます。

ローカライゼーションにサポートされた各言語のプロパティ ファイルが必要です。デフォルト以外の言語のファイルは、ファイル名の接尾辞で区別されます。次に例を示します。

my_resources.properties
my_resources_de.properties
my_resources_fr.properties

接尾辞は、言語コードの 2 つの文字の ISO 639-1 規格に従います。

プロパティ ファイルの各レコードには、次の形式があります。

id=text_string

ここでの意味は次の通りです。

id は、設定ファイルの buttonLabelKey、descriptionKey、dialogTitleKey、falseLabelKey、imagePathKey、labelKey、messageKey、textKey、または trueLabelKey 属性の識別子です。設定ファイルがどのように表示されるかによって、任意の接尾辞を使用したり、まったく使用しない場合がありますが、この識別子には通常、接尾辞 .LABEL があり、容易に区別することができます。

text_string は、項目のテキストです。

例:[ダイアログ] タブのローカライズ

ノード ダイアログのローカライズされたタブの例では、次の場所にある2つのプロパティ ファイル、デフォルト（英語）バージョンとフランス語のバージョンを使用します。

extension_folder**my_resources**.properties
extension_folder**my_resources**_fr.properties

この場合、extension_folder は、設定ファイルを含むフォルダです。

設定ファイルで、プロパティ ファイルは Resources セクションの Bundle 要素によって参照されます。

```
<Resources>
   <Bundle id="bundle" type="properties" path="my_resources"/>
</Resources>
```

path 属性には、言語拡張子または .properties 接尾辞を含むことはできません。

設定ファイルのその他の関連部分は次のとおりです。

```
<Node id="uiTest" scriptName="ui_test" type="dataTransformer" palette="recordOp" 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"/>
```

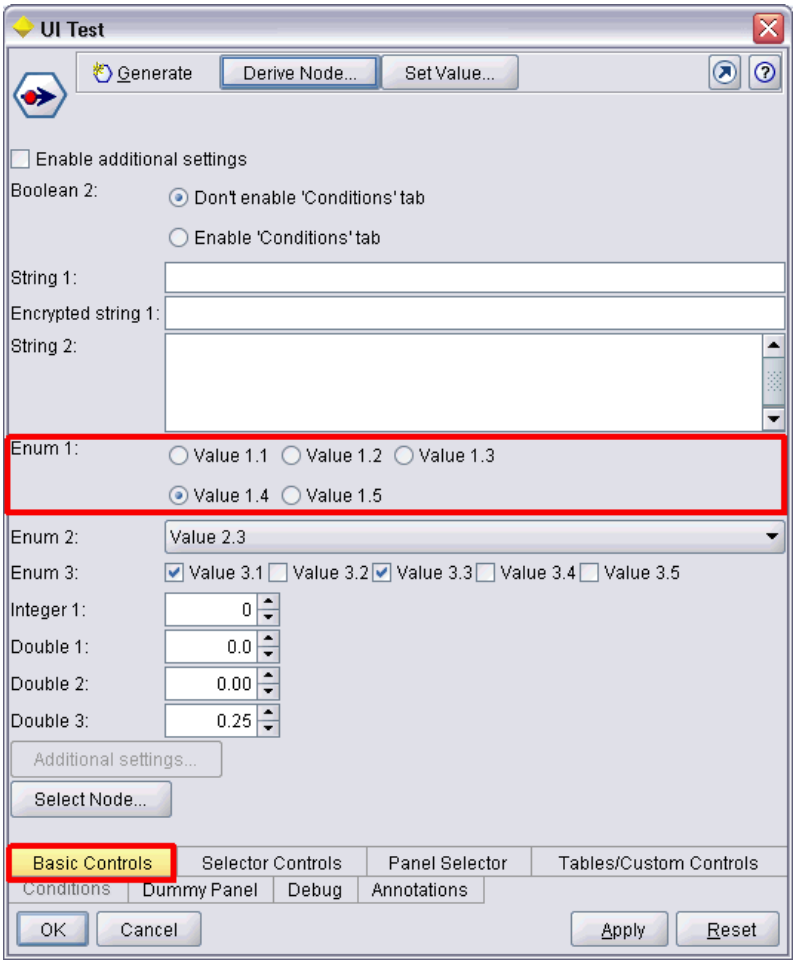
```
            <Enum value="value3" label="Value 1.3" labelKey="enum1.value3.LABEL"/>
            <Enum value="value4" label="Value 1.4" labelKey="enum1.value4.LABEL"/>
            <Enum value="value5" label="Value 1.5" labelKey="enum1.value5.LABEL"/>
        </Enumeration>
    </Property>
  </Properties>
...
  <UserInterface ...>
    <Tabs defaultTab="1">
      <Tab label="Basic Controls" labelKey="basicControlsTab.LABEL" ... >
   ...
  </UserInterface>
...
</Node>
```

プロパティ ファイルにおいて、プロパティ ファイルの英語バージョンには、次のレコードが含まれています。

```
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
```

これらのレコードに影響されたダイアログの次の一部が強調表示されます。

図 8-1
[非ローカライズ] タブ

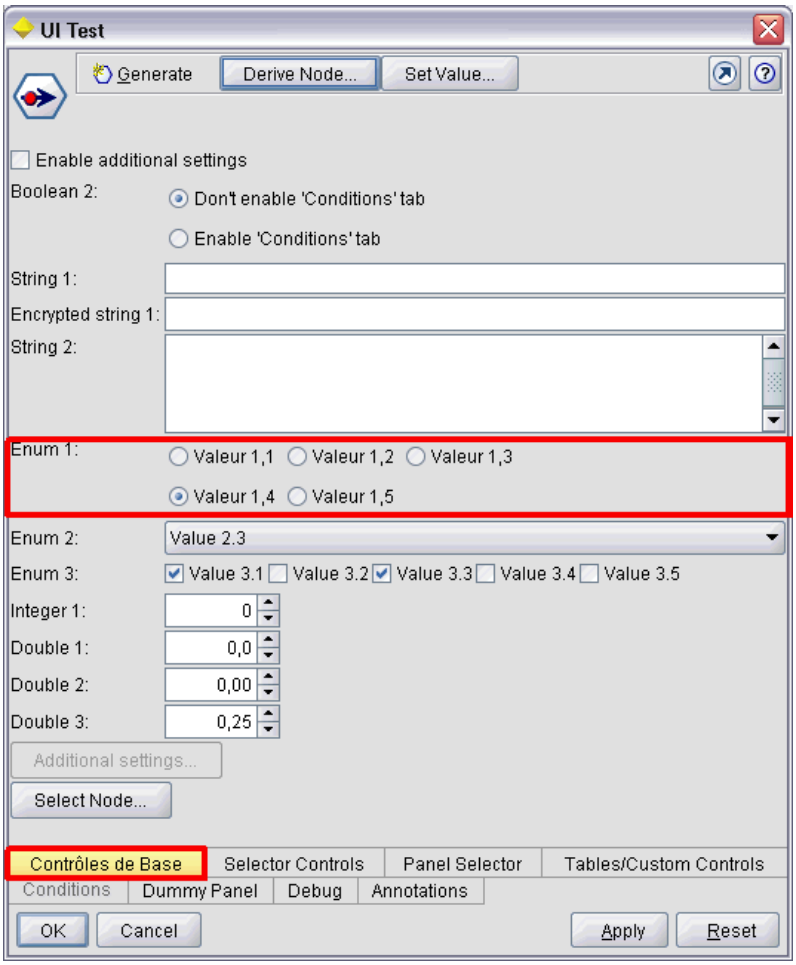


プロパティ ファイルのフランス語バージョン（my_resources_fr.properties）内の対応するセクションは、次のようになります。

```
basicControlsTab.LABEL=Contrôles 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
```

これらのレコードにより、画面の関連する部分に翻訳されたテキストが表示されます。

図 8-2
[ローカライズ] タブ



画面の下部の 4 つのボタンのローカライゼーションは、CLEF ではなく標準 IBM® SPSS® Modeler 機能によって操作されます。

例:特殊文字の使用

プロパティ ファイルでは、標準 ASCII テキスト文字列の特殊文字に Unicode 拡張シーケンスを使用する必要があります。次は、フランス語にローカライズされたプロパティ ファイルの一部です。

```
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
...
```

日本語または中国語などラテン文字以外の文字を使用する言語の場合、テキスト文字列に Unicode 拡張シーケンスを使用する必要があります。次は、日本語にローカライズされたレコードの同じセットです。

```
Genlinnode.LABEL=\u4e00\u822c\u5316\u7dda\u578b

Fields.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
...
```

ヘルプ ファイル

ヘルプ システムを持つ拡張子をローカライズする場合、ヘルプ システムのローカライズされたバージョンも提供する必要があります。それぞれのローカライズされた拡張子に対してローカライズされたヘルプ システムを提供します。

HTML Help のローカライズ

ローカライズしている拡張子が HTML Help ファイル（接尾辞 .chm）を使用する場合、デフォルトの .chm ファイルをローカライズされたバージョンと置き換えることができます。HTML Help システムの詳細は、「HTML Help」 (p.224). を参照してください。

ローカライズされた .chm ファイルを作成する手順は、次のとおりです。

- ヘルプ システムを構成し、同じファイル名を持つ個々のヘルプ トピック（.htm または .html）ファイルの翻訳されたバージョンを作成します。
- 必要であれば、ヘルプ システムに含まれるグラフィック（スクリーンショットなど）のローカライズされたバージョンを使用します。
- Microsoft の HTML Help Workshop または他のヘルプ オーサリング ツールを使用して、ファイルをローカライズされた .chm ファイルにコンパイルします。
- ローカライズされたノードを含むヘルプ システムを検証します。 詳細は、 p.237 ローカライズされた CLEF ノードの検証 を参照してください。

JavaHelp のローカライズ

ローカライズしている拡張子が JavaHelp システムを使用する場合、サポートされた各言語のヘルプ ソース ファイルのローカライズされたバージョンを提供する必要があります。JavaHelp では、存在する場合はローカライズされた適切なバージョンを表示できます。 詳細は、 7 章 p. 225 JavaHelp を参照してください。

ローカライズされた JavaHelp システムを作成する手順は、次のとおりです。

- ヘルプ システムを構成し、同じファイル名を持つ個々のヘルプ トピック（.htm または .html）ファイルの翻訳されたバージョンを作成します。
- 必要であれば、ヘルプ システムに含まれるグラフィック（スクリーンショットなど）のローカライズされたバージョンを使用します。
- ヘルプセットおよびその他の必要なファイル（マップ ファイル、コンテンツ、インデックス ファイル）を生成します。
- ローカライズされたノードを含むヘルプ システムを検定します。 詳細は、 p. 237 ローカライズされた CLEF ノードの検定 を参照してください。

ローカライズされた CLEF ノードの検定

ローカライズされたノードおよびヘルプ システムを検定する手順は次のとおりです。

- ローカライズされたノードの設定ファイルの Resources セクションで、HelpInfo 要素の path 属性を変更し、ローカライズされた .chm または .hs ファイルを参照します。たとえば HTML Help の場合、次のように使用します。

```
<Resources>
 ...
   <HelpInfo id="help" type="HTMLHelp" path="help/mynode_fr.chm "/>
</Resources>
```

JavaHelp の場合は、次のように使用します。

```
<Resources>
 ...
   <HelpInfo id="help" type="javahelp" helpset="help/mynode_fr.hs "/>
</Resources>
```

- ローカライズされた .chm または .jar ファイルを path 属性に示された場所にコピーします。

▶ 必要なロケールの Windows の地域を設定します。
コントロール パネル > 地域および言語のオプション > 地域のオプション > 標準および形式 > <言語>

▶ IBM® SPSS® Modeler を起動し、該当する言語で表示されるようにします。

▶ ローカライズされたノードを SPSS Modeler に追加します。 詳細は、10 章 p.274 CLEF 拡張のテスト を参照してください。

▶ ノードのコピーを領域に配置します。

ノードのダイアログを開き、該当する言語で正しく表示されていることを確認します。

▶ ダイアログの［ヘルプ］ボタンをクリックし、適切なヘルプ トピックが該当する言語で表示されていることを確認します。

アクセシビリティ

CLEF ノードでは、 マウス操作に代わるキーボード操作、画面読み上げソフトウェアのサポートなど、すべての標準的な IBM® SPSS® Modeler アクセス機能を利用できます。 詳細は、 A 付録 IBM SPSS Modeler のアクセス機能の概要 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

さらに、CLEF ノードでアクセスの目的でカスタマイズされたツールヒントのテキストを提供できます。

キーボードの組み合わせを指定して、エンド ユーザーに CLEF に追加されたさまざまなユーザー インターフェイス機能への代替アクセスを提供することもできます。 詳細は、 6 章 p.157 アクセス キーとキーボード ショートカット を参照してください。

操作ボタン、およびコントローラとして分類されるそれぞれの画面コンポーネント（チェック ボックスまたはラジオ ボタンのグループ）について、次のように定義できます。

- ラベル
- 説明

ラベルは、コンポーネントの名前として画面上に表示され、画面読み上げソフトウェアによって読み上げられる表示テキストです。視覚障害を持つユーザーの場合は、ラベルの表示フォント サイズを以下から取得できる［ユーザー オプション］ダイアログの［表示］タブのコントロールによって変更できます。
ツール > ユーザー オプション

説明は、マウス ポインタがコンポーネント上に移動した場合、表示されるツールヒントのテキストです。ツールヒントでは、名前だけで判断できないコンポーネントの詳細が提供されます。ツールヒントは、読み上げるように設定された画面読み上げソフトウェアによって読み上げられます。

ラベルおよび説明は、設定ファイルのコンポーネントを定義する label および description 属性によって定義されます。ラベルおよび説明のどちらも labelKey および descriptionKey 属性によってそれぞれローカライズできます。

例

この操作ボタンの例では、ラベルおよび説明機能のどちらの使用についても説明しています。

```
<Action id="setValue" label="値の設定..." labelKey="setValue.LABEL"
   description="値を設定" descriptionKey="setValue.TOOLTIP"/>
```

章 9

プログラミング

CLEF ノードのプログラミングについて

設定ファイルで定義できない処理を実行するノードを有効にするために、CLEF はプログラムで呼び出せる次のアプリケーション プログラミング インターフェース（APIs）を提供します。

- **クライアント側 API :**追加のコントロール、ユーザー インターフェイス コンポーネント、または設定ファイルで直接提供されない双方向性を必要とする拡張子で使用できる Java API です。
- **予測サーバー API (PSAPI) :**データ マイニングおよび予想分析機能を必要とするアプリケーションで使用するための IBM® SPSS® Modeler 機能を公開する Java API です。PSAPI および SPSS Modeler データ マイニング ワークベンチは同じ基本テクノロジーを共有しています。
- **サーバー側 API :**設定および実行設定の取得、これらの設定の持続、実行フィードバック、ジョブ制御（実行の中止など）、SQL 生成、返されたオブジェクトなどの側面をカバーする Cベースの APIです。

CLEF API ドキュメンテーション

次のセクションではクライアント側およびサーバー側の API の概要について説明します。さらに完全な API ドキュメントは IBM® SPSS® Modeler インストールにおける zip ファイルで含まれており、使用する前に抽出する必要があります。

API ドキュメントの抽出

▶ ¥Documentation¥en フォルダで製品 DVD のファイル clef_apidoc.zip を検索します。

▶ WinZip または同様のツールを用いて、zip ファイルの内容を適切なディレクトリへ抽出します。それによって、すべての APIドキュメントを含んだディレクトリ内の clef_apidoc サブフォルダが作成されます。

API ドキュメントの表示

▶ clef_apidoc サブフォルダへ移動して clef_apidoc.htm ファイルを開きます。

▶ 必要に応じて PSAPI のクライアント側またはサーバー側を選択します。

© Copyright IBM Corporation 1994, 2012.

クライアント側 API

CLEF はクライアント側の処理に使用できる方法を含めた多数の Java クラスを用意しています。たとえば DataModelProvider クラスは、入力データ モデルへの変更が複雑すぎて設定ファイルで提供される機能を使用できない場合、出力データの計算を行えるようにします。

クライアント側 API クラス

クライアント側のクラスは次のとおりです。

テーブル 9-1
クライアント側 API クラス

Class	説明
ActionHandler	拡張子でメニュー オプションやツールバー ボタンからユーザーがリクエストしたアクションを管理できます。
DataModelProvider	ノードで Java を使用してデータ モデルへの複雑な変更を行い、出力データ モデルを計算することができます。
ExtensionObjectFrame	モデルやドキュメント出力の表示に使用するウィンドウに関連した機能を定義します。
ExtensionObjectPanel	拡張オブジェクト パネルに関連した機能を定義します。
PropertyControl	プロパティ パネルのユーザー設定のプロパティ コントロールに関連した機能を定義します。

これらのクラスの詳細はクライアント側 API ドキュメントに記載されています。 詳細は、 p. 240 CLEF API ドキュメンテーション を参照してください。

クライアント側 API の使用方法

CLEF ノードにクライアント側関数の呼び出しを含むには、

- 関数呼び出しを含む .java ソース ファイルを作成します。
- ソース ファイルを .class ファイルへコンパイルします。
- 複数の .class ファイルを 1つの .jar ファイルにまとめ、設定ファイルにある .jar ファイルへの参照を含むことができます。−次に例を挙げます。

```
<Resources>
 ...
<JarFile id="selfjar" path="selflearning.jar"/>
 ...
```

```
</Resources>
```

CLEF 要素の中にはクラスを明示的に参照できるものもあります。たとえば次に示すように、クラス参照を設定ファイルの controlClass 要素の PropertyControl 属性にを含めることができます。

```
<PropertyControl property="target_field_values_specify" ...
   controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" ... />
```

CustomListControl はプロパティ コントロールを実装するクラスの名前、com.spss.clef.selflearning.propertycontrols.list は JarFile 要素で宣言した .jar ファイル内のクラスへのパスを示します。

詳細は、 4 章 p. 46 Resources セクション を参照してください。

このリリースで供給されているノード例のソース コードを見ると便利です。 詳細は、 3 章 p. 40 ソース コードの検証 を参照してください。

予測サーバー API (PSAPI)

PSAPI は基本となる予測サーバー技術へのプログラミング インターフェースを提供します。PSAPI の主な要素は Java インターフェースとして指定されています。これらインターフェースの大半が PSAPI の提供する内部クラスを通じて実装されますが、それは PSAPI 特性の一部を形成するものではありません。このアプローチは、予測サーバー技術になされた変更から PSAPI ユーザーを保護することを目的としています（アーキテクチャ変更、プライベート クライアントおよびサーバー プロトコルの変更など）。

これらのクラスの詳細は PSAPI ドキュメントに記載されています。 詳細は、 p. 240 CLEF API ドキュメンテーション を参照してください。

サーバー側 API

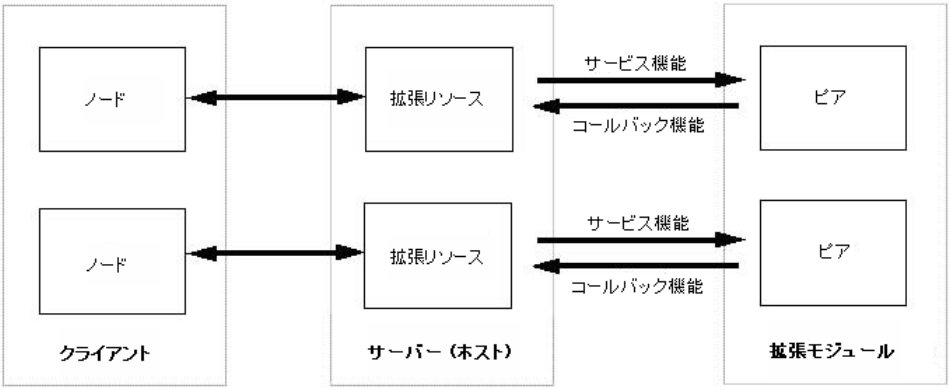
サーバー側 API はC 言語 API として定義されていますが、C++ の実装に対応します。拡張モジュールの開発者は、C または C++ でプログラミングすることにより、言語 API を無視して直接プログラミングすることを選択できます。開発者が C API へバインドする方法を持っていれば、他の言語も使用できます。CLEF はまた、一部の C API へのラッパー機能を果たす複数の C++ ヘルパー ソース ファイルを提供します。

アーキテクチャ

クライアントの拡張 **node** はサーバーの拡張 **peer** で補完されます。ピアは、サーバーがホストする共有ライブラリとして実装される、**拡張モジュール**によって定義されます。ノードとそのピア間のコミュニケーションは、サーバーが管理する拡張**リソース**によってメディエートされます。ピアがホストから情報がサービスをリクエストするために**コールバック関**

数を使用する一方で、リソースはそのピアを作成および操作するための拡張モジュールで定義される**サービス関数**を呼び出します。

図 9-1
CLEF API アーキテクチャ



サービス関数

サービス関数は拡張モジュールによって実装されます。拡張モジュールは必要とされるすべての関数を実装する必要があり、またそれ以外の一部またはすべてを実装する場合もあります。

サービス関数には 2 種類あります。

- モジュール関数
- ピア関数

次のセクションでは、サービス関数の概要について説明します。これらの関数に関する詳細は、次のようにサーバー側 API ドキュメントに記載されています。

▶ CLEF API ドキュメンテーション画面より、[**サーバー側 API の概要**] を選択します。

▶ [**モジュール**] タブをクリックします。

▶ [**拡張モジュールで実装する API サービス関数**] を選択します。

CLEF API ドキュメントへのアクセスに関する情報は、「CLEF API ドキュメンテーション」（ p.240 ）を参照してください。

モジュール関数

モジュール関数はシングル スレッドから呼び出されます。

関数	必須/任意	説明
clemext_initialise	はい	拡張モジュールとして初期化します。
clemext_cleanup	はい	拡張モジュールで割り当てられたリソースの処理
clemext_getModuleInformation	いいえ	拡張モジュールに関する情報を取得します
clemext_create_peer	はい	拡張ノードのピア インスタンスを作成します
clemext_destroy_peer	はい	ピア インスタンスの処理

ピア関数

ピア関数は clemext_create_peer への前の呼び出しで戻されたピア インスタンス ハンドルに適用されます。ピア関数は、ピア ハンドルが異なる場合のみ、別のスレッドから一斉に呼び出される場合があります。これに対する例外は、clemext_peer_cancelExecution 関数（定義される場合）が、別のスレッドでの長期間実行を中断するため常時どのスレッドからも呼び出される可能性があることです。

関数	必須/任意	説明
clemext_peer_configure	はい	ピア インスタンスにそのパラメータおよびデータ モデルを処理するよう指示します
clemext_peer_getDataModel	はい	ピア インスタンスから出力データ モデルを取得します
clemext_peer_getCatalogueInformation	いいえ	モジュールからカタログ情報を取得します
clemext_peer_getExecutionRequirements	いいえ	ピア インスタンスの実行要件を取得します
clemext_peer_beginExecution	はい	ピア インスタンスの実行を開始します
clemext_peer_nextRecord	はい	ピアの結果セットで次のレコードに移動します
clemext_peer_getRecordValue	はい	最後にフェッチした結果レコード内の特定フィールドの値を戻します
clemext_peer_endExecution	はい	ピア インスタンスに実行が完了したことを示します

関数	必須/任意	説明
clemext_peer_cancelExecution	いいえ	長時間の beginExecution または nextRecord の関数呼び出しを中断するための別のスレッドからの呼び出し
clemext_peer_getSQLGeneration	いいえ	ピア インスタンスから SQL を生成します
clemext_peer_getErrorDetail	はい	最後のピアにおけるモジュール固有のエラーに関する補足情報を取得します

次のピア関数はインタラクティブなモデル ビルダーで使用するために設計されています。

関数	必須/任意	説明
clemext_peer_beginInteraction	いいえ	ピア インスタンスとのインタラクションを開始します
clemext_peer_request	いいえ	ピアにおけるインタラクティブ リクエストを実行します
clemext_peer_getRequestReply	いいえ	最後に正常実行されたリクエストからの応答を取得します
clemext_peer_endInteraction	いいえ	ピア インスタンスにインタラクションが完了したことを示します

コールバック関数

拡張モジュールがホスト プロセスから情報またはサービスを要求するときは、**コールバック**を通じて行う必要があります。コールバックは、リクエストのターゲットを識別するポインターである**ハンドル**に適用されます。

コールバックは、呼び出しを指示する IBM® SPSS® Modeler オブジェクトのハンドルで渡すことにより起動します。ハンドルは、サービス関数のパラメータとして拡張モジュールへ渡されます。

コールバック関数に失敗すると、関連したモジュール固有のエラー コードで詳細を返す必要があります（CLEMEXTErrorCode に記載）。モジュールは代わりに、ホストが確認できるようコールバック エラーを戻しこの詳細を渡すことによって、これを管理できる場合があります。

次の種類の入力を利用することができます。

- ホスト関数
- ノード関数
- 反復関数

- 進行関数
- チャンネル関数（インタラクティブ モデル専用）

次のセクションでは、コールバック関数の概要について説明します。これらの関数に関する詳細は、次のようにサーバー側 API ドキュメントに記載されています。

▶ CLEF API ドキュメンテーション画面より、[**サーバー側 API の概要**] を選択します。

▶ [**モジュール**] タブをクリックします。

▶ [**一般コールバック**] を選択します。

CLEF API ドキュメントへのアクセスに関する情報は、「CLEF API ドキュメンテーション」（ p.240 ）を参照してください。

ホスト関数

ホスト関数は clemext_initialise から渡されるホスト ハンドルで定義されます。

関数	説明
clemext_host_getHostInformation	ホスト環境に関する統計情報を取得します

ノード関数

ノード関数は clemext_create_peer から渡されるホスト ハンドルで定義されます。

関数	説明
clemext_node_getNodeInformation	ノードに関する統計情報を取得します
clemext_node_getParameters	ノードからパラメータを取得します
clemext_node_getDataModel	ノードの入力データ モデルを取得します
clemext_node_getOutputDataModel	ノードの出力データ モデルを取得します
clemext_node_getSQLGeneration	ノードの上流 SQL 生成情報を取得します
clemext_node_getPassword	パスワード識別子の平文を取得します
clemext_node_getFilePath	実行中にクライアントとサーバーの間で交換されるファイルのパスを取得します

反復関数

反復関数は clemext_peer_beginExecution から渡される反復ハンドルで定義されます。反復は入力データセットを拡張モジュールへ公開します

関数	説明
clemext_iterator_nextRecord	入力データセットの次のレコードへ移動します
clemext_iterator_getRecordValue	最後にフェッチした入力レコード内の特定フィールドの値を戻します
clemext_iterator_rewind	nextRecord への次の呼び出しがセットの最初のレコードに移動するよう、入力データセットのステートを復元します

進行関数

進行関数は clemext_peer_beginExecution から渡される進行ハンドルで定義されます。

関数	説明
clemext_progress_report	進行状況をホストへ報告します

チャンネル関数

チャンネル関数はインタラクティブ モデルのみで使用し、clemext_peer_beginInteraction から渡されるチャンネル ハンドルで定義されます。

関数	説明
clemext_channel_send	クライアントへ非同期メッセージを送り、ピア スレッドが進行状況を報告できるようにします。

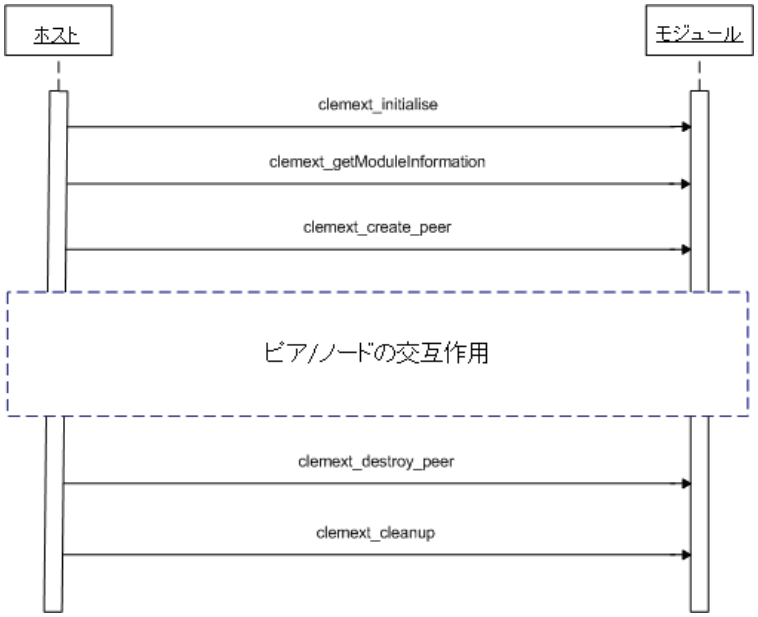
プロセス フロー

拡張モジュールは多数のサービスおよびコールバック関数を呼び出して、その処理を実行します。呼び出される実際の関数は、モジュールが実行するために必要とする処理によって異なります。

例

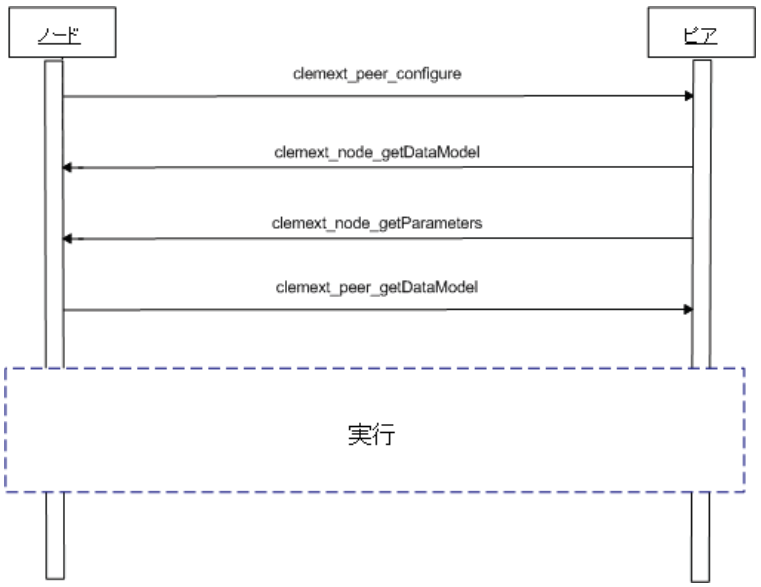
一般的なモジュール実行のシーケンス ダイアグラムは次のとおりです。

図 9-2
一般的なプロセス フロー



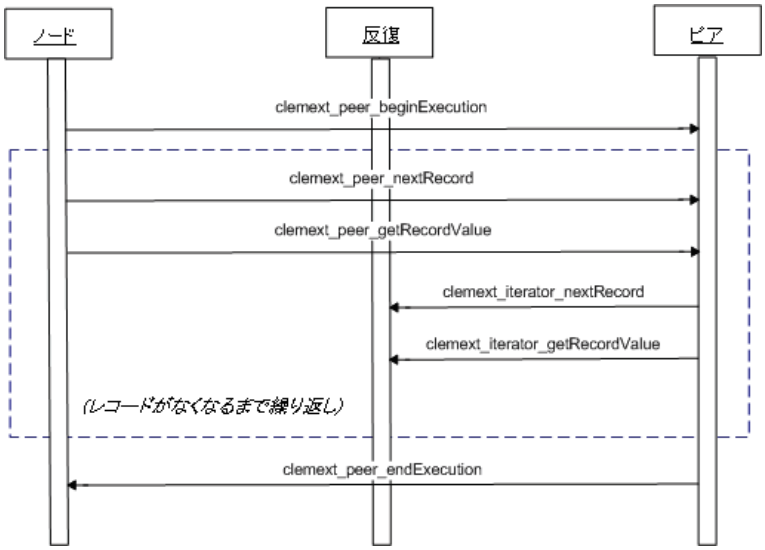
ピア/ノード インタラクション ブロックは次のようなものです。

図 9-3
一般的なピア/ノード インタラクション ブロック



一般的な実行ブロックは次のようなものです。

図 9-4
一般的な実行ブロック



注:

- モジュールは、サーバー開始時に IBM® SPSS® Modeler サーバー プロセスにロード、またはそのサービスが最初に要求されるときはオンデマンドで後にロードされる場合があります。
- モジュールがロードされると、ホストは サービス関数 clemext_initialise を一度起動します。
- モジュールが起動および初期化されると、ホストはサービス関数 clemext_getModuleInformation を使用してモジュールをクエリーする場合があります。
- モジュールがロードされた後、そのサービスはモジュールが提供するピア オブジェクトによって起動されます。モジュール内で、ピア オブジェクトはホストのノード オブジェクトの対としてサービス関数 clemext_create_peer によって作成され、ホスト アプリケーションによる指示にしたがってタスクの実行を管理します。同じタイプの複数ピア オブジェクトが存在し、それを一度のプロセスで同時に実行することが可能です。
- ピア オブジェクトが作成されると、サービス関数 clemext_peer_configure で設定できます。
- この時点で、clemext_node_getDataModel や clemext_node_getParameters など、クライアントから情報を得るためにコールバック関数を実行することがあります。
- SPSS Modelerは clemext_peer_getDataModel サービス関数を用いて、ピア インスタンスから出力データ モデルを取得します。

- ピア インスタンスの実行は clemext_peer_beginExecution サービス関数で開始します。
- clemext_peer_nextRecord サービス関数は、ピアの結果セットにおいてフォーカスを次のレコードに移動します（あるいは関数が初めて呼び出される場合は、最初のレコードに移動）。この後に現在のレコード内で特定フィールドの値を返す clemext_peer_getRecordValue サービス関数が続きます。
- 反復コールバック関数 clemext_iterator_nextRecord および clemext_iterator_getRecordValue は CLEF モジュールで呼び出し、入力レコードで並べて特定のフィールド値を返すことができます。
- ピア インスタンスの実行は clemext_peer_endExecution サービス関数で終了します。
- ピア インスタンスは clemext_destroy_peer を呼び出すと削除されます。
- モジュールがアンロードされる前に、ホストはサービス関数 clemext_cleanup を起動します。
- モジュールはサーバー プロセスがシャットダウンするとき、またはそれより早くサービスがもう必要なくなったときにアンロードされる場合があります。

サーバー側 API の機能

このセクションではサーバー側 API のいくつかの機能について取り上げます。

- ノード タイプの情報
- 異なるタイプのデータ ストレージを表すデータ タイプ
- サーバー側共有ライブラリ
- ファイルスペースおよび一時ファイル
- データベースでSQL 指示を実行する SQL プッシュバック
- IBM® SPSS® Modeler および拡張間のデータ モデル情報の交換
- 出力ドキュメント
- C++ ヘルパー

ノードの種類

設定ファイルにおいて、ノード定義は次の形式をとります。

```
<Node id="identifier" type="node_type" .../>
```

id 属性は独自にノードを識別する文字列です。

type 属性は次のタイプの一つとしてノードを識別します。

- データ リーダー
- データ ライター
- データ変換
- モデル ビルダー
- モデル アプライヤ
- ドキュメント ビルダー

詳細は、 2 章 p. 10 ノードの概要 を参照してください。

clemext_create_peer 関数には、id の値と Node の type 属性が引数として含まれます。

単一拡張モジュールは、ぞれぞれのタイプ内で、様々な機能を実行する様々なタイプのノードを実装します。たとえば、モジュールは次を実装する場合があります。

- データ ソースのためのデータ リーダーおよびデータ ライター
- 様々なモデル作成アルゴリズムのためのモデル ビルダーおよびモデル アプライヤ
- 様々なグラフ タイプに対応するドキュメント ビルダー

データおよびストレージ タイプ

ピア インスタンスは、実行開始時に提供される反復で clemext_iterator_getRecordValue を呼び出すことにより入力データを取得し、またホストからの clemext_peer_getRecordValue リクエストに応じて出力データを提供します。データはメモリでバイナリ形式で転送され、ピアおよびホストはデータ タイプを受け入れる必要があります。

バイナリ データ タイプはデータ モデルによって決定し、フィールドのストレージ属性に関連します。

次のテーブルでは、可能性のあるストレージ タイプがそれらを表すために使用するデータ タイプと一緒にリストされています。

テーブル 9-2
ストレージ タイプ

ストレージ タイプ	次によって表示	メモ
string	char *	文字列は常に UTF-8 コード化されています
real	CLEMEXTReal	倍精度浮動小数点数
integer	CLEMEXTInteger	64 ビット符号付きの整数
日付	CLEMEXTDate	1970 年 1 月 1 日からの日数を表す 64 ビット符号付きの整数
時間	CLEMEXTTime	午前零時からの秒数を表す 64 ビット符号付きの整数

ストレージ タイプ	次によって表示	メモ
timestamp	CLEMEXTTimestamp	1970 年 1 月 1 日からの秒数を表す 64 ビット符号付きの整数
unknown	–	表示できない不明のデータ タイプ–値を表します

ライブラリ

サーバー側の共有ライブラリは、設定ファイルでノード実行への対応を宣言できます。共有ライブラリのパスは、ホスト プロセスへ動的にロードされる共有ライブラリの場所を指定するために使用されます。共有ライブラリは必要な API 関数をすべて定義する必要があります。 詳細は、4 章 p.48 共有ライブラリ を参照してください。

モジュール名が設定ファイルで用意されている場合は（ノード定義の Execution セクション）、ピア オブジェクトを作成するために、その名前がサービス関数 clemext_create_peer の nodeId パラメータに渡されます。この場合、拡張は適切な種類のピア モジュールを作成できます。nodeType パラメータ値は作成されるピアの種類にも影響する場合があります。共有ライブラリが各タイプで1 つ以上のモジュールを実装しない可能性があるため、モジュール名は空白でもかまいません。

拡張モジュールを実装する共有ライブラリによって、依存ライブラリが必要な場合があります。これらは拡張共有ライブラリと同じディレクトリに存在しなくていはいけません。

一時ファイル

クライアント設定ファイルおよびサーバー拡張モジュールは、ピアが実行中に使用するファイルを作成できる一時的なプライベート スペースの filespace に関連したパス名を指定できます。ファイルスペースはピアのために作成されたサーバーの一時ディレクトリのサブディレクトリです。これは必要に応じて作成され、ピアが解除されると削除されます。

ピアはファイアスペースを存在時に完全制御します。ファイルスペースの完全パス名は**ノード情報ドキュメント**に記載されています。これは clemext_node_getNodeInformation コールバック関数の実行結果として返される XML 形式の情報です。 詳細は、 p.262 ノード情報ドキュメント を参照してください。

SQL プッシュバック

IBM® SPSS® Modeler ストリームが SQL データベースからデータを読み込んでデータ処理を実行する際、アドバンス ユーザーはデータベース自体で実行する SQL 指示をプッシュバックすることにより、この操作の効率性を向上できます。

いくつかの標準 SPSS Modeler ノードは SQL プッシュバックに対応しており、サーバー側 API には CLEF ノードも同様に可能にする関数呼び出しが含まれています。

clemext_peer_getSQLGeneration サービス関数はピア インスタンスから SQL を生成し、SQL 事項をデータベースへプッシュバックするために使用されます。データ リーダー ノードに関しては、生成された SQL はそれ自体でピアの結果セットを作成するに十分である必要があります。他のタイプのノードについては、作成された SQL は、ピアへの入力を提供する上流ノードのために生成された SQL へ依存する可能性が高いです。ピアはその関連したノード ハンドルで clemext_node_getSQLGeneration コールバック関数を呼び出すことにより、上流 SQL を取得できます。

データ モデルの処理

いくつかのサーバー側 API 呼び出しは、IBM® SPSS® Modeler と拡張モジュール間のデータ モデル情報の交換に関連しています。

- clemext_node_getDataModel はノードの入力データ モデルを取得します
- clemext_peer_getDataModel はピア インスタンスから出力データ モデルを取得します
- clemext_node_getOutputDataModel はノードの入力データ モデルを取得します

他の呼び出しはモジュール内外へのデータ渡しの方法と関連しています。データ モデルは、最後にフェッチした入力レコード内で指定されたフィールドの値を返す、次の関数でフィールド値を調べるために使用する索引を判断します。

- clemext_peer_getRecordValue
- clemext_iterator_getRecordValue

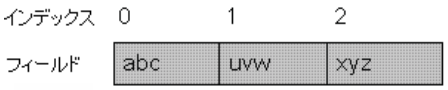
SPSS Modeler は clemext_node_getDataModel を呼び出して入力データ モデルのフィールドに関する情報を取得します。情報は次のように XML 形式で返されます。

```
<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>
```

モジュールはこの情報を使用して、次のように clemext_iterator_getRecordValue 関数を用いて入力レコードから数値を取得する際に、フィールド インデックスを提供することができます。

図 9-5
入力データ モデルの例



モジュールが入力データ モデルに影響する方法は、設定ファイルにおける OutputDataModel 要素の mode 属性の値によって制御されます。モジュールは次のことができます。

- 新規フィールドを追加してモデルを拡張する。
- 既存フィールドを削除または名前の変更をしてモデルを修正する。
- 既存モデルを新規フィールドと置き換える。
- モデルを変更せずそのままにしておく。

次の例はモデルの拡張および置換について説明しています。

例—入力データ モデルの拡張

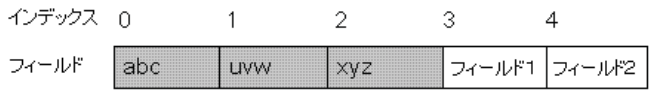
これは最もシンプルなケースで、モジュールで新規フィールドの追加や値の設定が可能になりますが、既存フィールドの値の削除や変更はできません。

設定ファイルにノード定義で次のような指示があると仮定します。

```
<OutputDataModel mode="extend">
 <AddField name="field1" storage="string" .../>
   <AddField name="field2" storage="real" .../>
   ...
</OutputDataModel>
```

ここでは、出力データ モデルが入力データ モデルにおけるすべてのフィールドを構成していると定義され、また 2 つの追加フィールドが OutputDataModel 要素で指定されています。したがって、出力データ モデルは 5 つのフィールドで構成されます。

図 9-6
入力データ モデルを拡張する出力データ モデルの例



clemext_peer_getDataModel 関数は、次のように追加されたフィールドに関する情報のみを返します。

```
<DataModel>
 <Fields>
 <Field name="field1" storage="string" .../>
       <Field name="field2" storage="real" .../>
   <Fields>
</DataModel>
```

この返される情報は設定ファイルの <AddField> 要素のタイプおよび数値（名前ではない）と一致する必要があります。

モジュールはコールバック関数 clemext_node_getOutputDataModel を使用して、IBM® SPSS® Modeler が追加を予測するフィールドの詳細を取得できます。この情報は clemext_peer_getDataModel への呼び出しに応じて SPSS Modeler へまっすぐ渡すことができます。これは、出力ファイルを作成および命名する設定ファイル ロジックが複雑な状況に便利です。

SPSS Modeler が clemext_peer_getRecordValue を呼び出す際、モジュールが各出力レコードの新しい値を提供します。新規フィールドのフィールド索引は入力フィールドの最後の索引の後に開始します。この例では、入力データ モデルが 3 つのフィールド（位置指数 0、1および2）を含んでいるため、2 つの出力フィールドは割り当てられたフィールド索引 3 および 4 です。モジュールがこれらのフィールドを変更できないため、SPSS Modeler は入力フィールドへ対応するフィールド索引で clemext_peer_getRecordValue を呼び出しません。

例—入力データ モデルの置換 (1)

この例では、拡張モジュールはその出力から入力データ モデル フィールドをすべて破棄し、新規フィールドに置き換えています。

設定ファイルには次が含まれています。

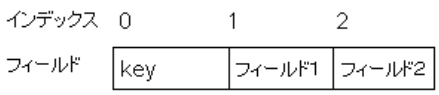
```
<OutputDataModel mode="modify">
 <AddField name="key" storage="integer" .../>
   <AddField name="field1" storage="real" .../>
   <AddField name="field2" storage="real" .../>
   ...
</OutputDataModel>
```

このとき、clemext_peer_getDataModel への呼び出しで返された XML データは、出力データ モデル内のすべてのフィールドについて説明しています。

```
<DataModel>
 <Fields>
 <Field name="key" storage="integer" .../>
       <Field name="field1" storage="real" .../>
       <Field name="field2" storage="real" .../>
    <Fields>
</DataModel>
```

出力データ モデルの構造は、次のようになっています。

図 9-7
入力データ モデルを置き換える出力データ モデルの例



clemext_peer_getRecordValue への呼び出しに使用されるフィールド索引は、最初の出力フィールド（key）が 0、次のフィールド（field1）が 1 という風に開始します。

例—入力データ モデルの置換 (2)

この例では、拡張によって提供される出力データ モデルもまた、前の例と同じく入力データ モデルを置き換えます。ただしこの場合は、出力データ モデルは設定ファイルで定義されておらず、代わりにサーバーの拡張モジュールによって実行時に計算されます。設定ファイルには次が含まれています。

```
<OutputDataModel mode="modify" method="sharedLibrary" libraryId="myLibraryId" />
```

出力データ モデルを計算するには、IBM® SPSS® Modeler が最初に clemext_peer_configure を呼び出し、次に clemext_peer_getDataModel が続きます。前の例にあるように、入力データ モデルのフィールドはどれも自動で出力データ モデルに含まれるのではなく、clemext_peer_getDataModel からのレスポンスで完全に定義されます。

注 ： このような場合、拡張モジュールがサーバーの出力データ モデルを定義するときは、「無効な操作」エラーをまねくため、モジュールは出力データ モデルを取得するために clemext_node_getOutputDataModel を使用できません。

XML 出力ドキュメント

サービスおよびコールバック関数の中には、XML 出力ドキュメントの形式でホストおよび拡張モジュール間の情報を転送するものがあります。多数の様々なドキュメントが入手できます。

XML 出力ドキュメント	メモ	次への呼び出しによる戻し
カタログ ドキュメント	カタログに関連するコントロールの値のリストを含みます。	clemext_peer_getCatalogInformation
データ モデル ドキュメント	ノードに入るまたはノード外のフィールドのセットについて説明します。	clemext_peer_getDataModel clemext_node_getDataModel clemext_node_getOutputDataModel
エラー詳細ドキュメント	エラーまたはその他の条件に関する情報を提供します。	clemext_peer_getErrorDetail
実行要件ドキュメント	ピア インスタンスに必要なデータ キャッシュや必須の入力フィールドなどの実行サポートについて説明します。	clemext_peer_getExecutionRequirements
ホスト情報ドキュメント	次のようなホスト環境に関する情報を提供します。製品識別子、説明、バージョン、プロバイダー、著作権、プラットフォームの詳細など。	clemext_host_getHostInformation
モジュール情報ドキュメント	次のような拡張モジュールに関する情報を提供します。モジュール識別子、説明、バージョン、プロバイダー、著作権、ライセンスの詳細など。	clemext_getModuleInformation
ノード情報ドキュメント	次のようなピア インスタンスに関するノードの情報を提供します。ノード識別子、タイプ、ファイルスペースの詳細など。	clemext_node_getNodeInformation
パラメータ ドキュメント	クライアント ノードからの設定パラメータを含みます。コンテンツは拡張子によって決定します。	clemext_node_getParameters
SQL 生成ドキュメント	ピアの実行がどのように SQL に翻訳できるかについて説明します。	clemext_peer_getSQLGeneration clemext_node_getSQLGeneration
状況詳細ドキュメント	実行中の進行および警告、または発生するその他の条件に関する補足情報を提供します。	clemext_progress_report

カタログ ドキュメント

カタログ ドキュメントは、UI コントロールから表示できる値のリストを含む、カタログの内容を説明します。

CLEF モジュールは、getCatalogInformation への呼び出しを次のように実行します。

```
CLEMEXTStatus
getCatalogInformation(
        const char *catalogId,
          char* buffer,
          size_t buffer_size,
          size_t* data_size,
          CLEMEXTErrorCode* errorCode) {
    …
}
```

ここで、catalogId は、設定ファイルの Catalog 要素で定義されている、特定のカタログの識別子です。

この関数は、カタログ ドキュメントを返します。

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>
```

データ モデル ドキュメント

データ モデル　ドキュメントは、ノード内またはノード外の**データ モデル**という、名前のついたフィールドのセット、および関連情報について説明します。データ型ノードで使用できる情報をカプセル化します。

入力（入力ノード）を使わないピアは空の入力モデルを持っており、また出力 8ターミナル ノード）を出さないピアは空の出力モデルを持っています。入力を使用して出力（プロセス ノード）を作成するピアは、その入力から出力モデルを計算する方法を知る必要があります。

ピアは関連するノード ハンドルの clemext_node_getDataModel を呼び出すことで、その入力データ モデルを取得できます。ピアはホストからの clemext_peer_getDataModel リクエストに応じてその出力データ モデルを提供します。

どのデータ モデルもそのプロパティと一緒にデータ モデル内のフィールドをすべて列挙するデータ ディクショナリとして直接表現できます。ノードによって提供されるピアに対する入力データ モデルは常にこの形式です。ピアで作成された出力データ モデルは同じ形式を持っている、または代わりに入力モデルに適用される操作（フィールドの追加、フィールドの削除、フィールドの修正）のシーケンスとして表現される場合があります。これは同じノードの出力モデルを大幅に簡略化します。

データ モデル ドキュメントのフィールドの順序は非常に重要で、対応する入力または出力データセットでどのデータを示すかという順番を決定します。

データ モデルは完全でなくデータの一部特性のみを提供する場合があります。ピアが実行プランを計算できるよう十分に指定された入力モデルは、そのピアに対して executable と呼びます。実行可能なデータ モデルは、入力および出力データが正確に配列できるよう、各フィールドに対するバイナリ タイプを常に含んでいる必要があります。

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>
```

エラー詳細ドキュメント

エラー詳細ドキュメントは IBM® SPSS® Modeler にメッセージ（エラー、警告、情報）を送り返すために使用され、エラーやその他の条件に関する情報を提供します。拡張モジュールはエラー詳細ドキュメントを提供し、ホストからの clemext_peer_getErrorDetail リクエストに応じてモジュール固有のエラーを説明することができます。

エラー詳細とは、各診断に少なくとも 1 つのエラー コード、メッセージ、そしてメッセージに挿入される詳細を含む 1 つ以上のパラメータのセットを含んだ 1 つ以上の Diagnostic 要素のセットです。エラー コードは設定ファイルの StatusCode 要素の値と一致します。

メッセージは異なる言語バリアントを持っている場合があり、そうでない場合はクライアントがエラー コードを使用してリソース バンドルからローカライズされたメッセージを選択できます。診断要素のシーケンスはエラーの因果連鎖を説明します。

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>
```

実行要件ドキュメント

実行要件ドキュメントはピア インスタンスに必要な実行サポートについて説明します。ピア インスタンスは、ホストからの clemext_peer_getExecutionRequirements リクエストに応じて実行要件ドキュメントを提供できます。正しい実行環境を提供するため、ホストはピアで clemext_peer_beginExecution を呼び出す前に、要件ドキュメントを参照します。

ホストは、モジュールが clemext_iterator_rewind 関数を使用することによって入力データに対して複数のパスができるよう、データ キャッシュ サービスを提供することができます。

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>
```

ホスト情報ドキュメント

ホスト情報ドキュメントはホスト環境について説明します。拡張モジュールはホスト ハンドルで clemext_host_getHostInformation を呼び出すことによりホスト情報を取得できます。

返される情報には製品識別子、詳細、バージョン、プロバイダー、著作権、プラットフォームなどが含まれます。

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>
```

モジュール情報ドキュメント

モジュール情報ドキュメントは拡張モジュールについて説明します。拡張モジュールは、ホストからの clemext_getModuleInformation リクエストに応じてモジュール情報ドキュメントを提供する必要があります。

返される情報にはモジュール識別子、詳細、バージョン、プロバイダー、著作権、ライセンスなどが含まれます。

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>
```

ノード情報ドキュメント

ノード情報ドキュメントはピア インスタンスに関連したノードについて説明します。ピア インスタンスはノード ハンドルの clemext_node_getNodeInformation を呼び出すことでノード情報を取得できます。ノード情報にはノード識別子、タイプ、ファイルスペースなどの詳細が含まれます。

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>
```

パラメータ ドキュメント

パラメータ ドキュメントには設定ファイルで定義された各 Property 要素の詳細が含まれます。詳細は、ピアがノードハンドルの clemext_node_getParameters を呼び出すことによって取得できる、設定パラメータの形式で返されます。

パラメータには名前と値があり、値は次のようにできます。

- 単純値（文字列）

- キー値（キーおよび値）
- 構造値（名前付き値のセット）
- 値のリスト

パラメータ ドキュメントの内容は拡張パッケージによって完全に決定します。パラメータはクライアント設定ファイルで定義され、サーバー拡張モジュールで解釈されます。

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 生成ドキュメント

SQL 生成ドキュメントは、ピアの実行がどのように SQL に翻訳できるかについて説明します。

ピアは、ホストからの clemext_peer_getSQLGeneration リクエストに応じて SQL 生成ドキュメントを提供できます。ホストは内部でのピア実行に優先して、SQLの実行を試みます。

入力を使用するピアは、関連するノード ハンドルの clemext_node_getSQLGeneration を呼び出すことで、その入力 SQL を取得できます。

SQL 生成ドキュメントの主なコンポーネントは、ノードまたはストリームのフラグメントの実行動作を複製する SQLステートメントです。データを作成するノードについては（つまり、データ リーダーまたはデータ トランスフォーマー ノード）、文は SELECT となる必要があり、また SELECT 文でコラム名にデータ モデルのフィールド名をマップするディクショナリを伴う必要があります。

SQL 生成ドキュメントは、データ ソース名や製品名など、ステートメントの実行に対するデータベース接続のプロパティも含んでいる場合があります。ピアはこれらのプロパティを使用して作成する SQL の決定を促すことができます。

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>
```

状況詳細ドキュメント

状況詳細ドキュメントは、進行状況や致命的でない警告、または実行中に発生するその他の条件に関する情報を提供します。拡張モジュールは、clemext_progress_report コールバック関数を用いて、状況詳細ドキュメントを非同期的に送ることができます。

状況詳細ドキュメントとは、各診断に少なくとも 1 つの条件 コード、メッセージ、そしてメッセージ（プロパティ ファイルに提供されていない場合）に挿入される詳細を含む 1 つ以上のパラメータのセットを含んだ 1 つ以上の Diagnostic 要素のセットで構成されています。StatusDetail 要素には、メッセージを次のいずれかに渡すよう指示するオプションの destination 属性もあります。

- IBM® SPSS® Modeler が管理するローカル トレース ファイル
- クライアント（ユーザーへ向けたメッセージ）
- すべて（可能性のあるすべての行き先に送る）

Diagnostic 要素の形式は次のとおりです。

```
<Diagnostic code="integer" severity="severity_level">
  <メッセージ>message_text</Message>
  <Parameter>value</Parameter>
</Diagnostic>
```

ここでの意味は次の通りです。

code（必須）は、条件コードを示す整数です。

severity は条件の重大度を示し、unknown、information、warning、error または fatal のいずれかとなります。

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>
```

ローカライズされたメッセージの使用

プロパティ ファイルのローカライズされたメッセージを使用する場合、状況詳細ドキュメントから Message 要素を省略し、次の例のように設定ファイルのメッセージ キーを使用します。

```
  ...
  <Execution ...>
  ...
    <StatusCodes>
      ...
<StatusCode code="21" status="warning" messageKey="fieldIgnoredMsg.LABEL"/>
      ...
    </StatusCodes>
  </Execution>
  ...
```

messages.properties ファイルには、次のものが含まれます。

```
fieldIgnoredMsg.LABEL=Field "{0}" はモデル作成に使用できず、無視されます。
```

状況詳細ドキュメントでは、フィールド名などのパラメータを送信して、次のようにローカライズされたメッセージに送信することができました。

```
<?xml version="1.0" encoding="utf-8"?>
<StatusDetail>
  <Diagnostic code="21">
    <Parameter>BP</Parameter>
  </Diagnostic>
</StatusDetail>
```

C++ ヘルパー

CLEF 例ノードのいくつかには、**helpers** として知られる事前定義済の C++ ソース ファイルが多く含まれています。これらはいくつかの C ベースのサーバー側 API にはラッパーとして働き、容易に C++ CLEF へコンパイルできます。

テーブル 9-3
C++ ヘルパー

ヘルパー	説明
BufferHelper	C API に使用されるサイズ変更が可能なメモリ バッファを管理します
DataHelper	データのラップ、読み書き操作に役立ちます。
HostHelper	LEMEXT HostHandle オブジェクトをラップします
XMLHelper	XML 処理をラップします

ヘルパーは .cpp および .h ファイルの一対の形式をとります。たとえば、BufferHelper.cpp および BufferHelper.h です。

これらのヘルパーファイルに関する詳細は、「ソース コードの検証」（ p.40 ）を参照してください。

これらのファイルに関する詳細は、次のようにサーバー側 API ドキュメントに記載されています。

- CLEF API ドキュメンテーション画面より、[**サーバー側 API の概要**] を選択します。
- [**ファイル**] タブをクリックします。
- 必要な情報のヘルパーに対応する .h ファイルの名前をクリックします。
- Data Structures の中で、対応するクラス名をクリックしてドキュメントを表示します。

CLEF API ドキュメントへのアクセスに関する情報は、「CLEF API ドキュメンテーション」（ p.240 ）を参照してください。

エラー処理

各関数呼び出しは、ステータス コード（CLEMEXTStatus）および オプションのモジュール固有エラー コード（CLEMEXTErrorCode）を返します。ステータス コードは success（エラーなし）、または API 関数用に列挙されたエラー コードの 1 つとなります。これらはほとんど常に「モジュール固有のエラー」を含みます。モジュール固有のエラー コードは、「モジュール固有エラーなし」を意味する 0 になる可能性もあります。

ステータス コード メッセージは IBM® SPSS® Modeler によって提供されます。一般的なステータス コードに関する詳細は、次のようにサーバー側 API ドキュメントに記載されています。

▶ CLEF API ドキュメンテーション画面より、[**サーバー側 API の概要**] を選択します。

▶ [**モジュール**] タブをクリックします。

▶ [**一般的なステータス コード**] を選択します。

CLEF API ドキュメントへのアクセスに関する情報は、「CLEF API ドキュメンテーション」（ p. 240 ）を参照してください。

モジュール固有エラー メッセージは次のように提供されることがあります。

- 設定ファイルで（モジュール セクションの StatusCodes で）
- 設定ファイルで参照されるリソース バンドルで
- 拡張モジュールによって

モジュール固有のエラー コードについては、モジュールが、デフォルトのエラー メッセージ（設定ファイルまたはリソース バンドルで説明されないエラー）およびメッセージに挿入されるパラメータから成る追加のエラー詳細を提供できます。複数のエラーメッセージはエラーの因果連鎖を説明できます。

クライアントでのエラー報告には次の形式があります。

```
node_label:message
```

ここでの意味は次の通りです。

- node_label はモジュールが指定される Node 要素の label 属性の値です。
- message は、サーバーから提供される、または設定ファイル（またはローカリゼーション用の .properties ファイル）で定義されるメッセージのテキストです。

XML API の解析

IBM® SPSS® Modeler には Apache の Xerces-C XML parser を含まれており、モジュールで XML データの読み書きができる多数のコールバックを提供します。希望により自身の XML パーサーで代用することもできます。

サーバー側 API の使用方法

ノードにサーバー側関数の呼び出しを含むには、

- 関数呼び出しを含む C++ .cpp and .h ソース ファイルを作成します。
- ソース ファイルをダイナミック リンク ライブラリ（.dll）ファイルへコンパイルします。
- 次のように、設定ファイルから .dll ファイルへの参照を含みます。

```
<Resources>
...
   <SharedLibrary id="mylib1" path="mycorp.mynode/mylib" />
.
</Resources>
```

詳細は、 4 章 p. 48 共有ライブラリ を参照してください。
このリリースで供給されているノード例のソース コードを見ると便利です。 詳細は、 3 章 p. 40 ソース コードの検証 を参照してください。

サーバー側のプログラミング ガイドライン

CLEF モジュールのサーバー側ダイナミック リンク ライブラリ（DLL）の部分は、モジュールが適切に機能し、IBM® SPSS® Modeler の操作の影響を回避するよう、ガイドラインに従う必要があります。CLEF モジュールは、次のことが必要です。

- ピア実行が内蔵されている
- 単一プロセスの複数のピアインスタンスをサポートする
- スレッドが安全であること
- スレッドまたはプロセス環境の警告を回避する
- モジュール内のスレッドの使用を制限する
- 実行キャンセルの要求を正しく処理する
- 中断されたシステムの呼び出しを再開する（UNIX）
- CoInitialize または CoUninitialize の呼び出し時に注意する（Windows）
- モジュールがアンロードされた場合の仮説を立てないようにする

- サブプロセス開始時に注意する
- 標準出力または標準誤差への書き込みを回避する

次のセクションでは、これらの領域についてより詳細に説明します。

ピア実行が内蔵されている

ピア インスタンスは IBM® SPSS® Modeler サーバー プロセス内の他のピア インスタンスの有無に関する推定はできません。SPSS Modeler は、ストリーム内で隣接するノードに対応するピア インスタンスが異なる段階で実際に実行されるよう、実行のスケジュールを設定できます。そのため、インスタンスの有無と実行は重複しません。

ピア インスタンスは内蔵されており、パイプまたはソケットなどを使用してそのほかのインスタンスと直接通信することはありません。ピア インスタンス間のすべての通信は、データを読み込んだり書き込んだりして、または外部エージェント（ピア間のデータ共有を管理するデータベース サーバーなど）を使用して直接実行する必要があります。

単一プロセスの複数のピアインスタンスをサポートする

エンド ユーザーは、ストリーム実行時のサーバー プロセスで、特定の CLEF モジュールで複数のピア インスタンス（同じタイプの複数のノードなど）を作成する場合があります。そのため、CLEF モジュールの静的データは複数のピア インスタンス間で共有され、ピア オブジェクト専用のデータを保存するために使用することはできません。静的データの例としては、C++ クラスの静的メンバー、C コンパイル ユニットのグローバルまたは静的変数があります。

CLEF モジュールの API 関数を再度使用し、再使用しないシステム コールを行わないようにする必要があります。たとえば、ピア インスタンスが clemext_iterator_nextRecord を使用して入力反復の入力データをフェッチすると、最初のピアの上流にあり、最初のピアに最終的に使用されるデータを作成する 2 番目のピア インスタンスに clemext_peer_nextRecord を呼び出します。

strtok のようなシステム コールは、再使用ではなく、また使用できません。再使用である代替の詳細は、ご使用のプラットフォームのオペレーティング システム マニュアルを参照してください。

スレッドを安全にする

IBM® SPSS® Modeler は、複数のピア インスタンスの実行を異なる実行スレッドからインターリーブする場合があります。そのため、ミューテックス（相互排除オブジェクト）または同様のスレッド ライブラリ サービス

と同期するなどして、ピア オブジェクト間で共有されたリソースへのアクセスを保護する必要があります。

CLEF モジュールは、スレッドセーフでないシステム コールを行わないようにする必要があります。詳細は、ご使用のオペレーティング システムのマニュアル、または UNIX man を参照してください。

スレッドまたはプロセス環境の警告を回避する

スレッドまたはプロセスを呼び出す環境が変更されるシステム コールを使用しないようにします。

そのような呼び出しの例は、次のとおりです。ただし、このリストは完全ではありません。

- setlocale : ロケール情報の読み取りではなくロケールの変更に使用する場合
- SetCurrentDirectory (Windows) または chdir (UNIX)
- LogonUser (Windows) または seteuid (UNIX)
- putenv
- exit
- signal

注 :Windows の場合、CoInitialize は、スレッドの環境を変更しますが必要な場合があります。 詳細は、 p.271 CoInitialize または CoUninitialize のコール時に注意する (Windows) を参照してください。

モジュール内のスレッドの使用を制限する

通常、モジュールはスレッドを内部で自由に使用できます。ただし、コールバック関数は、IBM® SPSS® Modeler が CLEF モジュール関数 (clemext_peer_cancelExecution 以外) を呼び出すために使用したスレッドのみに起動する必要があります。

次のコールバック関数はモジュール内で実行しているスレッドから非同期的に呼び出すことができます。

- clemext_progress_report
- clemext_channel_send

ピア インスタンスは、複数のスレッドがこれらの呼び出しのそれぞれに同時に起動しないようにする必要があります。

実行キャンセルの要求を正しく処理する

エンド ユーザーがピア インスタンスの実行のキャンセルを要求した場合、IBM® SPSS® Modeler はモジュールの clemext_peer_cancelExecution 関数への非同期的な呼び出しを実行します。開発者は、この呼び出しを実行する必要があります。この関数は非同期でコールされ、別の CLEF API 関数コールが実行されている間にコールされます。

中断されたシステムの呼び出しを再開する (UNIX)

UNIX の場合、IBM® SPSS® Modeler アプリケーションは信号および信号ハンドラを使用します。一部の UNIX システム コールは、コール実行時に信号を受信した場合に EINTR コードを返す場合があります。特定の UNIX プラットフォームのシステム コールについては、man ページを参照してください。

このイベントが発生すると、コールしたコードは EINTR リターン コードを確認し、コールを再開する必要があります。これを実行する方法の 1 つとして、簡単なラッパー関数（open_safe）を作成して、アプリケーションが該当するラッパーをコールするようにします。

```
int
open_safe(const char* path, int oflag, mode_t mode) {
 int res;
 while ((res = ::open(path, oflag, mode)) == -1
                        && errno == EINTR) {
 }
 return res;
}
```

CoInitialize または CoUninitialize のコール時に注意する (Windows)

Windows の場合、Windows Component Object Model (COM) ライブラリ サービスを使用する必要のあるスレッドは、COM サービス使用前にシステム API 関数 CoInitialize をコールし、完了時に CoUninitialize をコールする必要があります。IBM® SPSS® Modeler がモジュールに CLEF API を起動するスレッドは、CoInitialize をコールする場合もあれば、コールしない場合もあります。

これらのスレッドから COM サービスを使用する CLEF モジュールは、通常 clemext_create_peer または clemext_peer_beginExecution 関数で CoInitialize をコールする必要があります。コールが成功した場合、スレッドで実行が完了したときに、CoUninitialize を通常 clemext_destroy_peer または clemext_peer_endExecution でコールする必要もあります。

CoInitiaize コールの詳細は、Microsoft Developer Network (MSDN) の Web サイト (http://msdn.microsoft.com) にあるマニュアルを参照してください。

モジュールがアンロードされた場合の仮説を立てないようにする

現在、CLEF モジュールはセッション終了までロードされたままになっています (モジュールを必要に応じてアンロードおよび再ロードすることはできません)。モジュールがロードされる IBM® SPSS® Modeler サーバー プロセスからの終了時にも、関数 clemext_cleanup はコールされません。そのため、開発者はいかなる場合にも、モジュールがアンロードされ、リソースが解放されるという仮説を立てることはできません。

サブプロセス開始時に注意する

サブプロセスを開始すると、CreateProcess (Windows) または fork (UNIX) によって、親プロセスおよび子プロセスの対話、および親にオープンなリソースを子プロセスが継承する状況に多くの混乱をきたす場合があります。

CLEF モジュールがプロセス外で実行する必要がある場合、適切な代替のアーキテクチャの使用を検討してください。たとえば、CLEF モジュールは、必要なタスクを実行するアプリケーション サーバーに提供されたサービスを使用する場合があります。

特に、Windows のプロセスでは、TRUE に設定された bInheritHandles パラメータを含む CreateProcess 関数を使用してサブプロセスを開始しないようにする必要があります。そうすることによって、子プロセスは、親 (IBM® SPSS® Modeler サーバー) プロセスでオープンなすべてのファイル記述子を継承します。

標準出力または標準誤差への書き込みを回避する

CLEF モジュールがプロセスの標準出力または標準誤差のストリームに書き込むと (おそらくデバッグの目的)、通常エンド ユーザーには表示されません。ただし、CLEF ノードを含むストリームが IBM® SPSS® Modeler Solution Publisher を使用して表示され、コマンド ライン シェルから実行されると (Windows または UNIX)、この出力が表示され、ユーザーは混乱する場合があります。

代わりに、CLEF モジュールは、ホスト コールバック関数 clemext_host_trace をコールして文字列形式で表示するメッセージを渡して、トレース サービスを起動することができます。IBM® SPSS® Modeler Server 構成オプション ファイル (IBM® SPSS® Modeler インストール ディレクトリの /config/options.cfg) で次の設定を使用し、SPSS Modeler のインストールでトレースを有効にする必要もあります。

trace_extension, 1

トレースされたメッセージは、SPSS Modeler インストール ディレクトリのファイル /log/trace-<process_ID>-<process_ID>.log に出力されます。process_ID は、SPSS Modeler Server プロセスの識別子です。セッションはすべて同じログ ファイルを共有しているため、複数のセッションを同時にトレースしないようにしてください。

章
10

テストと配布

CLEF 拡張のテスト

他のユーザーに配布する前に新しい拡張をテストすることをお勧めします。

特性ファイルおよび関連するリソース バンドル、.jar ファイル、共有ライブラリおよびユーザー ヘルプ ファイルを作成した後、ファイルを必須ファイル構造に配置し、それらをローカルの IBM® SPSS® Modeler のインストール ディレクトリにコピーすることによって拡張をテストできます。次回 SPSS Modeler を起動するときに、新しい拡張が SPSS Modeler ユーザー インターフェイスに表示されます。

CLEF 拡張のテスト

- IBM® SPSS® Modeler が開いている場合は終了します。
- 拡張が CLEF ノードまたは出力を定義する場合、拡張が正しく機能するまで、拡張ダイアログの［デバッグ］タブを有効にすることをお勧めします。 詳細は、 p.276［デバッグ］タブの使用 を参照してください。
- クライアント側ファイルおよびサーバー側ファイルを必須構造に配置します。特性ファイルおよびノードで必要な関連したリソース（.jar または .dll ファイルなど）が適切な場所にコピーされます。 詳細は、 1 章 p.6 ファイル構造 を参照してください。
- クライアント側のディレクトリを SPSS Modeler インストール ディレクトリの ¥ext¥lib フォルダにコピーします。
- サーバー側のディレクトリを SPSS Modeler インストール ディレクトリの ¥ext¥bin フォルダにコピーします。
- SPSS Modeler を起動します。
- 拡張でメニューまたはメニュー項目を定義する場合、メイン メニュー システムに正しく表示されていることを確認します。拡張が新しいノードを定義する場合、特性ファイルに定義されているように、正しいノード パレットの該当する位置にノードが表示されていることを確認します。
- 拡張を徹底的にテストします。

  たとえば次のことを確認します。

© Copyright IBM Corporation 1994, 2012.

- フィールドおよびレコードの数が増えてもノードのパフォーマンスが低下していない
- ヌル値が一貫して処理されている
- 必要に応じてさまざまなロケール（ヨーロッパ、極東）がサポートされている

▶ 拡張を定義した後でも、特性ファイルに変更を行うことができます。ただし、変更は SPSS Modeler を再起動するまでは無効です。

CLEF 拡張のデバッグ

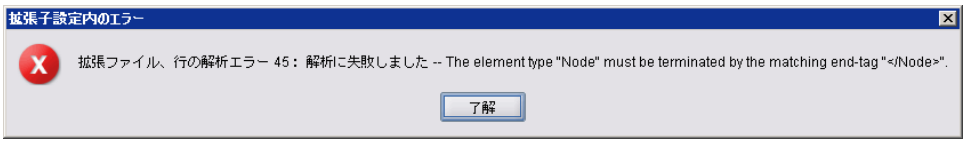
CLEF では、拡張のデバッグを支援する次の機能を定期要します。

- XML 構文エラーのメッセージ
- 外部実行
- ［デバッグ］タブ

XML 構文エラー

特性ファイルの不正な XML 構文は、XML 構文解析のエラー メッセージでフラグが付けられます。たとえば、次のとおりです。

図 10-1
XML 構文エラー



メッセージでは、エラーの内容とともに、エラーのおおよその行番号を表示します。

この状況を解決する手順は次のとおりです。

▶ ファイルのエラーを修正します。

▶ CLEF 拡張のテスト p.274 の手順に従って、ファイルを再テストします。

▶ 特性ファイルに構文エラーがなくなるまで、この手順を繰り返します。

外部実行

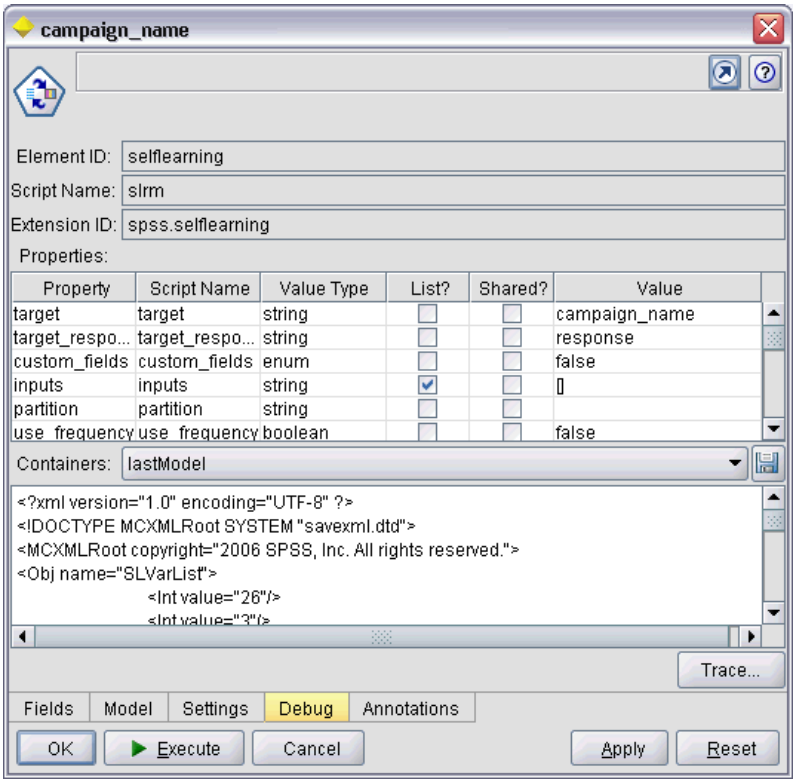
通常、ユーザーによって記述された CLEF 拡張は、IBM® SPSS® Modeler プロセスとは別に独自のプロセスで実行します。これにより、デバッグ時、拡張プロセスが失敗した場合でも、IBM® SPSS® Modeler Server プロセス全体に取り込まれることがありません。

注 ： デフォルト設定を無効にすることはできません。 詳細は、 p. 277 実行オプションの変更 を参照してください。

[デバッグ] タブの使用

CLEF ノードまたは出力に関連するダイアログまたはフレームの場合、[デバッグ] タブを有効化してオブジェクトのプロパティ設定を調査できるようにします。また、拡張で定義されたコンテナの内容を表示して、これらの内容をさらに調査するためにファイルに保存することもできます。 詳細は、 4 章 p. 73 コンテナ を参照してください。

図 10-2
ダイアログの [デバッグ] タブ



[デバッグ] タブを有効にするには、特性ファイルの Extension 要素の debug 属性の値を true に設定します。 詳細は、 4 章 p. 45 Extension 要素 を参照してください。

タブのフィールドは、次のとおりです。

要素 ID: 拡張の一意の識別子で、特性ファイルの ExtensionDetails 要素の id 属性の値です。

スクリプト名:スクリプトで参照された場合の一意の識別子で、Node 要素の scriptName 属性の値です。

拡張 ID:拡張のファイルとディレクトリ リソースが存在する拡張フォルダの名前です。値は、ExtensionDetails 要素の providerTag 属性および id 属性（「.」文字によって区切り）を連結することによって形成されています。識別子の providerTag の部分には、値には文字列 spss が含まれ、内部の使用に指定されます。

プロパティ:この表では、ノードの Property 宣言の選択された詳細情報を示しています。

- **プロパティ**:プロパティの一意の識別子で、Property 要素の name フィールドの値です。
- **スクリプト名**:スクリプトで参照された場合のプロパティの一意の識別子で、Property 要素の scriptName 属性の値です。
- **値のデータ型**:Property 要素の valueType 属性で定義された、このプロパティがとることのできる値のデータ型です。
- **リスト?**: プロパティが指定された値のデータ型の値のリストかどうかを示す、Property 要素の isList 属性の値です。
- **共有しますか？** チェックすると、このプロパティが拡張内の複数の場所（モデル ビルダー ノード、モデル出力、モデル アプライヤ）で使用されていることを示します。
- **値**:プロパティのデフォルト値です（ある場合）。

コンテナ:選択されたコンテナの内容（モデル データなど）を表示しますこのフィールドをクリックして、拡張に定義されたその他のコンテナのリストを表示し、さまざまなコンテナを選択して内容を表示します。隣接する［**コンテナの保存**］ボタンをクリックして、より詳細に調査するために選択されたコンテナの内容を XML 形式で保存します。

トレース:ノードが実行される場合にトレース 出力ができるようにするダイアログを表示します。

実行オプションの変更

デフォルトでは、ユーザーによって記述された CLEF 拡張モジュールは、IBM® SPSS® Modeler プロセスとは別のプロセスで実行します。このように、実行プロセスの失敗によって、SPSS Modeler プロセスが失敗することはありません。これに対し、IBM Corp. が提供するモジュールは、デフォルトではメインのプロセスで実行します。

2 つのサーバー構成オプションによって、システム管理者は指定されたモジュールのこれらのケースを反対のケースに変更することができます。2 つのオプションは、モジュール識別子のカンマで区切られたリストで、変更の影響を受けているモジュールを示します。

注 : これらのオプションのいずれかの変更は、通常、カスタマ サポート担当者の要求によってのみ実行できます。

オプションは次の通りです。

プロセス内実行オプション

このオプションで、通常外部プロセスにロードされる拡張モジュール（通常、ユーザーによって記述されたモジュール）を SPSS Modeler に直接ロードすることができます。形式は次のとおりです。

clef_inprocess_execution, "moduleID1[,moduleID2[,...moduleIDn]]"

この場合、moduleID は、関連する特性ファイルの ExtensionDetails 要素の id 属性の値です。次に例を示します。

clef_inprocess_execution, "test.example_filereader"

外部実行オプション

このオプションで、通常 SPSS Modeler に直接ロードされる拡張モジュール（通常、IBM Corp. によって提供されたモジュール）を外部プロセスにロードすることができます。形式は次のとおりです。

clef_external_execution, "moduleID1[,moduleID2[,...moduleIDn]]"

ここで、moduleID は、clef_inprocess_execution の場合と同じです。架空の例を次に示します。

clef_external_execution, "spss.naivebayes,spss.terminator"

実行オプションの追加または変更

実行オプションを追加または変更するには、『SPSS Modeler 15 サーバー管理およびパフォーマンス ガイド』の「options.cfg ファイルの使用」に示された手順に従います。

CLEF 拡張の配布

新しい拡張がすべてテストされると、配布の準備ができます。

- ［デバッグ］タブが有効な場合は、無効にします。 詳細は、 p. 276［デバッグ］タブの使用 を参照してください。
- 拡張ファイルをインストールする方法に正確に反映されるファイル構造を作成します。 詳細は、 1 章 p. 6 ファイル構造 を参照してください。
- ファイル構造を .zip ファイルに圧縮します。クライアント側インストールおよびサーバー側インストールにそれぞれ .zip ファイルを作成するをより簡単です。
- .zip ファイルをエンド ユーザーに配布します。

CLEF 拡張のインストール

CLEF 拡張をインストールする手順は、次のとおりです。

- 拡張ファイル構造を .zip ファイルを受け取った後、IBM® SPSS® Modeler インストール ディレクトリの \ext\lib フォルダにクライアント側のファイルを展開します。
- SPSS Modeler インストール ディレクトリ（IBM® SPSS® Modeler Server を使用する場合同等のディレクトリ）の \ext\bin フォルダにサーバー側のファイルを展開します。
- IBM® SPSS® Modeler 起動して、新しいノードがノード パレットの該当する場所に表示されていることを確認します。

CLEF 拡張のアンインストール

CLEF 拡張をアンインストールする手順は、次のとおりです。

- IBM® SPSS® Modeler インストール ディレクトリの \ext\lib ディレクトリにある拡張フォルダを検索します。

  拡張でサーバー側の拡張フォルダもインストールしている場合、SPSS Modeler または IBM® SPSS® Modeler Server ディレクトリの \ext\bin ディレクトリでこのフォルダを検索します。
- 拡張フォルダを削除します。

  変更は、次回 IBM® SPSS® Modeler を起動した場合に有効になります。

付録

A

CLEF XML スキーマ

CLEF 要素の参照

このセクションでは、CLEF のすべての要素の参照を提供しています。

各トピックには、要素および親要素や子要素の有効な属性を示しています。図にすべての要素の子を表示しています。図内の矢印は、要素間で共有することができる要素を示しています。これらの要素は、親トピックの子としてではなく、このトピック（「CLEF 要素の参照」）の子として目次に表示されています。

Action 要素

テーブル A-1
の属性 Action

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string
id	**必須の**		string
imagePath	optional		string
imagePathKey	optional		string
label	**必須の**		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>
```

© Copyright IBM Corporation 1994, 2012.

</xs:element>

親 要素

Actions 要素

ActionButton 要素

テーブル A-2
の属性 ActionButton

属性	使用	説明	有効値
action	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

Actions 要素

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>
```

親 要素

CommonObjects 要素

子 要素

Action 要素

AddField 要素

テーブル A-3
の属性 AddField

属性	使用	説明	有効値
direction	オプション		**in** **out** **both** **none** **partition**
directionRef	optional		
fieldRef	optional		
group	optional		string
label	オプション		string
missingValuesRef	optional		
name	***必須の***		
prefix	optional		string

属性	使用	説明	有効値
role	optional		**unknown** **predictedValue** **predictedDisplay-Value** **probability** **residual** **standardError** **entityId** **entityAffinity** **upperConfidence-Limit** **lowerConfidenceLimit** **propensity** **value** **supplementary**
storage	オプション		**unknown** **integer** **real** **string** **date** **time** **timestamp**
storageRef	optional		
tag	optional		string
targetField	optional		string
type	オプション		**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>
```

親 要素

ForEach 要素, ModelFields 要素

子 要素

MissingValues 要素, ModelField 要素, NumericInfo 要素, Range 要素, Values 要素

関連 要素

ChangeField 要素

MissingValues 要素

テーブル A-4
の属性 MissingValues

属性	使用	説明	有効値
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>
```

親 要素

AddField 要素

子 要素

Range 要素, Values 要素

ModelField 要素

テーブル A-5
の属性 ModelField

属性	使用	説明	有効値
group	オプション		
role	**必須の**		**unknown** **predictedValue** **predictedDisplay-Value** **probability** **residual** **standardError** **entityId** **entityAffinity** **upperConfidence-Limit** **lowerConfidenceLimit** **propensity** **value** **supplementary**
tag	オプション		string
targetField	オプション		string
value	オプション		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>
```

親 要素

AddField 要素

And 要素

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>
```

親 要素

Command 要素, Constraint 要素, CreateDocument 要素, CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModel 要素, CreateModelApplier 要素, CreateModelOutput 要素, Enabled 要素, Not 要素, Option 要素, Or 要素, Run 要素, Visible 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

Attribute 要素

テーブル A-6
の属性 Attribute

属性	使用	説明	有効値
defaultValue	optional		
description	optional		string
descriptionKey	optional		string
isList	optional		boolean
label	**必須の**		string
labelKey	optional		string
name	**必須の**		string
valueType	オプション		**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>
```

親 要素

Catalog 要素, Structure 要素

BinaryFormat 要素

XML 表示

```
<xs:element name="BinaryFormat"></xs:element>
```

親 要素

FileFormatType 要素

Catalog 要素

テーブル A-7
の属性 Catalog

属性	使用	説明	有効値
id	**必須の**		string
valueColumn	**必須の**		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>
```

親 要素

Catalogs 要素

子 要素

Attribute 要素

Catalogs 要素

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>
```

親 要素

CommonObjects 要素

子 要素

Catalog 要素

ChangeField 要素

テーブル A-8
の属性 ChangeField

属性	使用	説明	有効値
direction	オプション		**in** **out** **both** **none** **partition**
directionRef	optional		
fieldRef	**必須の**		
label	オプション		string
missingValuesRef	optional		
name	**必須の**		

属性	使用	説明	有効値
storage	オプション		**unknown** **integer** **real** **string** **date** **time** **timestamp**
storageRef	optional		
type	オプション		**auto** **range** **discrete** **set** **orderedSet** **flag** **typeless**
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: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>
```

親 要素

ForEach 要素, ModelFields 要素

子 要素

MissingValues 要素, ModelField 要素, NumericInfo 要素, Range 要素, Values 要素

関連 要素

AddField 要素

MissingValues 要素

テーブル A-9
の属性 MissingValues

属性	使用	説明	有効値
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>
```

親 要素

ChangeField 要素

子 要素

Range 要素, Values 要素

ModelField 要素

テーブル A-10
の属性 ModelField

属性	使用	説明	有効値
group	オプション		
role	**必須の**		**unknown** **predictedValue** **predictedDisplay-Value** **probability** **residual** **standardError** **entityId** **entityAffinity** **upperConfidence-Limit** **lowerConfidenceLimit** **propensity** **value** **supplementary**
tag	オプション		string
targetField	オプション		string
value	オプション		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>
```

親 要素

ChangeField 要素

CheckBoxControl 要素

テーブル A-11
の属性 CheckBoxControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

CheckBoxGroupControl 要素

テーブル A-12
の属性 CheckBoxGroupControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

ClientDirectoryChooserControl 要素

テーブル A-13
の属性 ClientDirectoryChooserControl

属性	使用	説明	有効値
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	**必須の**		**open** **save** **import** **export**
property	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

ClientFileChooserControl 要素

テーブル A-14
の属性 ClientFileChooserControl

属性	使用	説明	有効値
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	**必須の**		**open** **save** **import** **export**
property	**必須の**		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: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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

ComboBoxControl 要素

テーブル A-15
の属性 ComboBoxControl

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean

属性	使用	説明	有効値
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	**必須の**		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>
```

テーブル A-16
拡張 タイプ

タイプ	説明
ItemChooserControl	

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

Command 要素

テーブル A-17
の属性 Command

属性	使用	説明	有効値
path	**必須の**		

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>
```

親 要素

Run 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

CommonObjects 要素

Provides a location for definitions that are global to the extension

テーブル A-18
の属性 CommonObjects

属性	使用	説明	有効値
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>
```

親 要素

Extension 要素

子 要素

Actions 要素, Catalogs 要素, ContainerTypes 要素, FileFormatTypes 要素, PropertySets 要素, PropertyTypes 要素

Condition 要素

テーブル A-19
の属性 Condition

属性	使用	説明	有効値
container	optional		string
control	optional		string
expression	optional		
listMode	optional		all any
op	必須の		equals notEquals isEmpty isNotEmpty lessThan lessOrEquals greaterThan greaterOrEquals equalsIgnoreCase isSubstring startsWith endsWith startsWithIgnoreCase endsWithIgnoreCase isSubstring hasSubstring isSubstringIgnore-Case hasSubstringIgnore-Case in countEquals countLessThan countLessOrEquals countGreaterThan countGreaterOrE-quals contains storageEquals

属性	使用	説明	有効値
			typeEquals directionEquals isMeasureDiscrete isMeasureContinuous isMeasureTypeless isMeasureUnknown isStorageString isStorageNumeric isStorageDatetime isStorageUnknown isModelOutput modelOutputRoleE–quals modelOutputTarget–FieldEquals modelOutputHas–Value modelOutputTagE–quals 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: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>
```

親 要素

And 要素, Command 要素, Constraint 要素, CreateDocument 要素, CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModel 要素, CreateModelApplier 要素, CreateModelOutput 要素, Enabled 要素, ExpertSettings 要素, Not 要素, Option 要素, Or 要素, Run 要素, Visible 要素

Constraint 要素

テーブル A-20
の属性 Constraint

属性	使用	説明	有効値
property	**必須の**		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>
```

親 要素

AutoModeling 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

Constructors 要素

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>
```

親 要素

DocumentOutput 要素, Execution 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

子 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

Container 要素

テーブル A-21
の属性 Container

属性	使用	説明	有効値
name	**必須の**		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>
```

親 要素

Containers 要素, Containers 要素, Containers 要素, Containers 要素, Containers 要素

ContainerFile 要素

テーブル A-22
の属性 ContainerFile

属性	使用	説明	有効値
container	optional		
containerType	optional		
path	**必須の**		

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>
```

親 要素

InputFiles 要素, OutputFiles 要素

ContainerTypes 要素

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>
```

親 要素

CommonObjects 要素

子 要素

DocumentType 要素, ModelType 要素

Controls 要素

XML 表示

```
<xs:element name="Controls">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
            <xs:element ref="Menu"></xs:element>
            <xs:element ref="MenuItem"></xs:element>
            <xs:element ref="ToolbarItem"></xs:element>
        </xs:choice>
    </xs:sequence>
</xs:element>
```

親 要素

UserInterface 要素

子 要素

Menu 要素, MenuItem 要素, ToolbarItem 要素

CreateDocument 要素

テーブル A-23
の属性 CreateDocument

属性	使用	説明	有効値
sourceFile	**必須の**		string
target	**必須の**		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>
```

親 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

関連 要素

CreateModel 要素

CreateDocumentOutput 要素

テーブル A-24
の属性 CreateDocumentOutput

属性	使用	説明	有効値
type	**必須の**		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>
```

親 要素

Constructors 要素

子 要素

And 要素, Condition 要素, CreateDocument 要素, CreateModel 要素, Not 要素, Or 要素, SetContainer 要素, SetProperty 要素

関連 要素

CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

CreateInteractiveDocumentBuilder 要素

テーブル A-25
の属性 CreateInteractiveDocumentBuilder

属性	使用	説明	有効値
type	**必須の**		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>
```

親 要素

Constructors 要素

子 要素

And 要素, Condition 要素, CreateDocument 要素, CreateModel 要素, Not 要素, Or 要素, SetContainer 要素, SetProperty 要素

関連 要素

CreateDocumentOutput 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

CreateInteractiveModelBuilder 要素

テーブル A-26
の属性 CreateInteractiveModelBuilder

属性	使用	説明	有効値
type	**必須の**		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: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>
```

親 要素

Constructors 要素

子 要素

And 要素, Condition 要素, CreateDocument 要素, CreateModel 要素, Not 要素, Or 要素, SetContainer 要素, SetProperty 要素

関連 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

CreateModel 要素

テーブル A-27
の属性 CreateModel

属性	使用	説明	有効値
signatureFile	optional		string
sourceFile	**必須の**		string
target	**必須の**		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>
```

親 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

子 要素

And 要素, Condition 要素, ModelDetail 要素, Not 要素, Or 要素

関連 要素

CreateDocument 要素

ModelDetail 要素

テーブル A-28
の属性 ModelDetail

属性	使用	説明	有効値
algorithm	**必須の**		string

XML 表示

```
<xs:element name="ModelDetail" maxOccurs="unbounded">
    <xs:attribute name="algorithm" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

CreateModel 要素

CreateModelApplier 要素

テーブル A-29
の属性 CreateModelApplier

属性	使用	説明	有効値
type	**必須の**		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: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>
```

親 要素

Constructors 要素

子 要素

And 要素, Condition 要素, CreateDocument 要素, CreateModel 要素, Not 要素, Or 要素, SetContainer 要素, SetProperty 要素

関連 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelOutput 要素

CreateModelOutput 要素

テーブル A-30
の属性 CreateModelOutput

属性	使用	説明	有効値
type	**必須の**		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:element ref="CreateDocument"></xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

Constructors 要素

子 要素

And 要素, Condition 要素, CreateDocument 要素, CreateModel 要素, Not 要素, Or 要素, SetContainer 要素, SetProperty 要素

関連 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素

DatabaseConnectionValue 要素

A value specifying the details of a database connection.

テーブル A-31
の属性 DatabaseConnectionValue

属性	使用	説明	有効値
connectionType	**必須の**		string
datasourceName	**必須の**		string
password	**必須の**		string
userName	**必須の**		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>
```

親 要素

ListValue 要素, Attribute 要素, ListValue 要素, Parameter 要素, Attribute 要素, ListValue 要素

DataFile 要素

テーブル A-32
の属性 DataFile

属性	使用	説明	有効値
path	**必須の**		

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>
```

親 要素

InputFiles 要素, OutputFiles 要素

子 要素

DelimitedDataFormat 要素

DataFormat 要素

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>
```

親 要素

FileFormatType 要素

子 要素

DelimitedDataFormat 要素, SPSSDataFormat 要素

DataModel 要素

The data model coming into or out of a node. An input/Output data model is a set of Fieldsl.

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 要素

FieldFormats 要素, FieldGroups 要素, Fields 要素

FieldFormats 要素

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.

テーブル A-33
の属性 FieldFormats

属性	使用	説明	有効値
count	オプション		nonNegativeInteger
defaultNumberFormat	**必須の**		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>
```

親 要素

DataModel 要素

子 要素

NumberFormat 要素

NumberFormat 要素

Defines format information for a numeric field.

テーブル A-34
の属性 NumberFormat

属性	使用	説明	有効値
decimalPlaces	**必須の**		nonNegativeInteger
decimalSymbol	**必須の**		**period** **comma**

属性	使用	説明	有効値
formatType	**必須の**		**standard** **scientific** **currency**
groupingSymbol	**必須の**		**none** **period** **comma** **space**
name	**必須の**		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>
```

親 要素

FieldFormats 要素

FieldGroups 要素

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.

テーブル A-35
の属性 FieldGroups

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger

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>
```

親 要素

DataModel 要素

子 要素

FieldGroup 要素

FieldGroup 要素

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”.

テーブル A-36
の属性 FieldGroup

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger
countedValue	オプション		string
displayLabel	オプション		string
groupType	**必須の**		**fieldGroup** **multiCategorySet** **multiDichotomySet**
name	**必須の**		

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>
```

親 要素

FieldGroups 要素

子 要素

FieldName 要素

FieldName 要素

テーブル A-37
の属性 FieldName

属性	使用	説明	有効値
name	**必須の**		

XML 表示

```
<xs:element name="FieldName">
    <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

親 要素

FieldGroup 要素

Fields 要素

テーブル A-38
の属性 Fields

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger

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>
```

親 要素

DataModel 要素

子 要素

Field 要素

Field 要素

テーブル A–39
の属性 Field

属性	使用	説明	有効値
direction	オプション		**in** **out** **both** **none** **partition**
displayLabel	オプション		string
name	**必須の**		string

属性	使用	説明	有効値
storage	オプション		**unknown** **integer** **real** **string** **date** **time** **timestamp**
type	オプション		**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: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:attribute name="name" type="xs:string" use="required"></xs:attribute>
    <xs:attribute name="type" type="FIELD-TYPE" default="auto">
        <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="storage" type="FIELD-STORAGE" default="unknown">
        <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="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>
```

親 要素

Fields 要素

子 要素

DisplayLabel 要素, MissingValues 要素, Range 要素, Values 要素

DBConnectionChooserControl 要素

テーブル A-40
の属性 DBConnectionChooserControl

属性	使用	説明	有効値
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	**必須の**		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="DBConnectionChooserControl">
    <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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

DBTableChooserControl 要素

テーブル A-41
の属性 DBTableChooserControl

属性	使用	説明	有効値
connectionProperty	**必須の**		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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

DefaultValue 要素

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>
```

親 要素

Property 要素, PropertyType 要素

子 要素

Identifier 要素, ServerTempDir 要素, ServerTempFile 要素

ServerTempFile 要素

テーブル A-42
の属性 ServerTempFile

属性	使用	説明	有効値
basename	**必須の**		

XML 表示

```
<xs:element name="ServerTempFile">
    <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

親 要素

DefaultValue 要素

ServerTempDir 要素

テーブル A-43
の属性 ServerTempDir

属性	使用	説明	有効値
basename	**必須の**		

XML 表示

```
<xs:element name="ServerTempDir">
    <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

親 要素

DefaultValue 要素

Identifier 要素

テーブル A-44
の属性 Identifier

属性	使用	説明	有効値
basename	**必須の**		

XML 表示

```
<xs:element name="Identifier">
    <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

親 要素

DefaultValue 要素

DelimitedDataFormat 要素

テーブル A-45
の属性 DelimitedDataFormat

属性	使用	説明	有効値
delimeter	optional		**tab** **comma** **semicolon** **colon** **verticalBar** **other**
eol	optional		**cr** **crlf** **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="delimeter" 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="crlf"></xs:enumeration>
                <xs:enumeration value="lf"></xs:enumeration>
                <xs:enumeration value="other"></xs:enumeration>
            </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>
```

親 要素

DataFile 要素, DataFormat 要素

DisplayLabel 要素

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.

テーブル A-46
の属性 DisplayLabel

属性	使用	説明	有効値
lang	オプション		NMTOKEN
value	**必須の**		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>
```

親 要素

Field 要素

DocumentBuilder 要素

XML 表示

```
<xs:element name="DocumentBuilder">
    <xs:sequence>
```

```
        <xs:element name="DocumentGeneration"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

Node 要素

子 要素

DocumentGeneration 要素

DocumentGeneration 要素

テーブル A-47
の属性 DocumentGeneration

属性	使用	説明	有効値
controlsId	**必須の**		string

XML 表示

```
<xs:element name="DocumentGeneration">
    <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

DocumentBuilder 要素

関連 要素

ModelingFields 要素, ModelGeneration 要素, ModelEvaluation 要素

DocumentOutput 要素

テーブル A-48
の属性 DocumentOutput

属性	使用	説明	有効値
deprecatedScriptNames	optional		string
id	**必須の**		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>
```

親 要素

Extension 要素

子 要素

Constructors 要素, Containers 要素, ModelProvider 要素, Properties 要素, UserInterface 要素

関連 要素

InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

Containers 要素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Container"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

DocumentOutput 要素

子 要素

Container 要素

DocumentType 要素

Defines a new document type

テーブル A-49
の属性 DocumentType

属性	使用	説明	有効値
format	**必須の**		**utf8** **binary**
id	**必須の**		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>
```

親 要素

ContainerTypes 要素

関連 要素

ModelType 要素

Enabled 要素

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>
```

親 要素

ActionButton 要素, CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, ComboBoxControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, MultiFieldChooserControl 要素, MultiItemChooserControl 要素, PasswordBoxControl 要素, PropertiesPanel 要素, PropertiesSubPanel 要素, PropertyControl 要素, RadioButtonGroupControl 要素, SelectorPanel 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SingleItemChooserControl 要素, SpinnerControl 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素, TextBrowserPanel 要素, ItemChooserControl タイプ

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

Enumeration 要素

XML 表示

```
<xs:element name="Enumeration">
    <xs:sequence>
        <xs:element name="Enum" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

PropertyType 要素

子 要素

Enum 要素

Enum 要素

テーブル A-50
の属性 Enum

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string
imagePath	optional		string
imagePathKey	optional		string
label	**必須の**		string
labelKey	optional		string
value	**必須の**		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>
```

親 要素

Enumeration 要素

ErrorDetail 要素

Suuplementary 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 要素

Diagnostic 要素

Diagnostic 要素

テーブル A-51
の属性 Diagnostic

属性	使用	説明	有効値
code	**必須の**		integer
severity	オプション		**unknown** **information** **warning** **error** **fatal**
source	オプション		string
subCode	オプション		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>
```

親 要素

ErrorDetail 要素

子 要素

Message 要素, Parameter 要素

Message 要素

テーブル A-52
の属性 Message

属性	使用	説明	有効値
lang	オプション		NMTOKEN

XML 表示

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
    <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>
```

親 要素

Diagnostic 要素

Parameter 要素

XML 表示

```
<xs:element name="Parameter" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>
```

親 要素

Diagnostic 要素

Executable 要素

XML 表示

```
<xs:element name="Executable">
    <xs:sequence>
        <xs:element ref="Run" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

Execution 要素

子 要素

Run 要素

Execution 要素

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>
```

親 要素

Node 要素

子 要素

Constructors 要素, Executable 要素, InputFiles 要素, Module 要素, OutputFiles 要素, Properties 要素

Extension 要素

Defines the top-level extension container.

テーブル A-53
の属性 Extension

属性	使用	説明	有効値
debug	optional		boolean
version	**必須の**		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 要素

CommonObjects 要素, DocumentOutput 要素, ExtensionDetails 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, License 要素, ModelOutput 要素, Node 要素, Resources 要素, UserInterface 要素

ExtensionDetails 要素

Defines information about the extension such as the extension id, the extension provider and version information.

テーブル A-54
の属性 ExtensionDetails

属性	使用	説明	有効値
copyright	optional		string
description	optional		string
id	**必須の**		string
label	**必須の**		string
provider	optional		string
providerTag	**必須の**		string
version	オプション		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>
```

親 要素

Extension 要素

ExtensionObjectPanel 要素

テーブル A-55
の属性 ExtensionObjectPanel

属性	使用	説明	有効値
id	optional		string
panelClass	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素, Tab 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

Field 要素

テーブル A-56
の属性 Field

属性	使用	説明	有効値
direction	オプション		**in** **out** **both** **none** **partition**
label	オプション		string
name	**必須の**		

属性	使用	説明	有効値
storage	オプション		**unknown** **integer** **real** **string** **date** **time** **timestamp**
type	オプション		**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 要素

MissingValues 要素, ModelField 要素, NumericInfo 要素, Range 要素, Values 要素

MissingValues 要素

テーブル A-57
の属性 MissingValues

属性	使用	説明	有効値
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>
```

親 要素

Field 要素

子 要素

Range 要素, Values 要素

ModelField 要素

テーブル A-58
の属性 ModelField

属性	使用	説明	有効値
group	オプション		
role	**必須の**		**unknown** **predictedValue** **predictedDisplay-Value** **probability** **residual** **standardError** **entityId** **entityAffinity** **upperConfidence-Limit** **lowerConfidenceLimit** **propensity** **value** **supplementary**
tag	オプション		string
targetField	オプション		string
value	オプション		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>
```

親 要素

Field 要素

FieldFormats 要素

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.

テーブル A-59
の属性 FieldFormats

属性	使用	説明	有効値
count	オプション		nonNegativeInteger
defaultNumberFormat	**必須の**		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 要素

NumberFormat 要素

NumberFormat 要素

Defines format information for a numeric field.

テーブル A-60
の属性 NumberFormat

属性	使用	説明	有効値
decimalPlaces	**必須の**		nonNegativeInteger
decimalSymbol	**必須の**		**period** **comma**
formatType	**必須の**		**standard** **scientific** **currency**
groupingSymbol	**必須の**		**none** **period** **comma** **space**
name	**必須の**		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>
```

親 要素

FieldFormats 要素

FieldGroup 要素

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”.

テーブル A-61
の属性 FieldGroup

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger
countedValue	オプション		string
displayLabel	オプション		string
groupType	**必須の**		**fieldGroup multiCategorySet multiDichotomySet**
name	**必須の**		

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 要素

FieldName 要素

FieldName 要素

テーブル A-62
の属性 FieldName

属性	使用	説明	有効値
name	**必須の**		

XML 表示

```
<xs:element name="FieldName">
    <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

親 要素

FieldGroup 要素

FieldGroups 要素

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.

テーブル A-63
の属性 FieldGroups

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger

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 要素

FieldGroup 要素

FieldGroup 要素

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”.

テーブル A-64
の属性 FieldGroup

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger
countedValue	オプション		string
displayLabel	オプション		string
groupType	**必須の**		**fieldGroup** **multiCategorySet** **multiDichotomySet**
name	**必須の**		

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>
```

親 要素

FieldGroups 要素

子 要素

FieldName 要素

FieldName 要素

テーブル A-65
の属性 FieldName

属性	使用	説明	有効値
name	**必須の**		

XML 表示

```
<xs:element name="FieldName">
    <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

親 要素

FieldGroup 要素

FileFormatType 要素

テーブル A-66
の属性 FileFormatType

属性	使用	説明	有効値
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>
```

親 要素

FileFormatTypes 要素

子 要素

BinaryFormat 要素, DataFormat 要素, UTF8Format 要素

FileFormatTypes 要素

XML 表示

```
<xs:element name="FileFormatTypes">
    <xs:sequence>
        <xs:element ref="FileFormatType" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

CommonObjects 要素

子 要素

FileFormatType 要素

ForEach 要素

テーブル A-67
の属性 ForEach

属性	使用	説明	有効値
container	optional		string
from	optional		string
inFields	optional		string
inFieldValues	optional		string
inProperty	optional		string
step	optional		string

属性	使用	説明	有効値
to	optional		string
var	**必須の**		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>
```

親 要素

ModelFields 要素

子 要素

AddField 要素, ChangeField 要素, ForEach 要素, RemoveField 要素

Icon 要素

テーブル A-68
の属性 Icon

属性	使用	説明	有効値
imagePath	**必須の**		string
resourceID	optional		string
type	**必須の**		**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>
```

親 要素

Icons 要素, Palette 要素

Icons 要素

XML 表示

```
<xs:element name="Icons">
    <xs:sequence>
        <xs:element ref="Icon" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

UserInterface 要素

子 要素

Icon 要素

InputFiles 要素

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>
```

親 要素

Execution 要素, Module 要素

子 要素

ContainerFile 要素, DataFile 要素

InteractiveDocumentBuilder 要素

テーブル A-69
の属性 InteractiveDocumentBuilder

属性	使用	説明	有効値
deprecatedScriptNames	optional		string
id	**必須の**		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: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>
```

親 要素

Extension 要素

子 要素

Constructors 要素, Containers 要素, ModelProvider 要素, Properties 要素, UserInterface 要素

関連 要素

DocumentOutput 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

Containers 要素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Container"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

InteractiveDocumentBuilder 要素

子 要素

Container 要素

InteractiveModelBuilder 要素

テーブル A-70
の属性 InteractiveModelBuilder

属性	使用	説明	有効値
deprecatedScriptNames	optional		string
id	**必須の**		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>
```

親 要素

Extension 要素

子 要素

Constructors 要素, Containers 要素, ModelProvider 要素, Properties 要素, UserInterface 要素

関連 要素

DocumentOutput 要素, InteractiveDocumentBuilder 要素, ModelOutput 要素, Node 要素

Containers 要素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Container"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

InteractiveModelBuilder 要素

子 要素

Container 要素

Layout 要素

テーブル A-71
の属性 Layout

属性	使用	説明	有効値
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>
```

親 要素

ActionButton 要素, CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, ComboBoxControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, MultiFieldChooserControl 要素, MultiItemChooserControl 要素, PasswordBoxControl 要素, PropertiesPanel 要素, PropertiesSubPanel 要素, PropertyControl 要素, RadioButtonGroupControl 要素, SelectorPanel 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SingleItemChooserControl 要素, SpinnerControl 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素, TextBrowserPanel 要素, ItemChooserControl タイプ

子 要素

Cell 要素

Cell 要素

テーブル A-72
の属性 Cell

属性	使用	説明	有効値
column	**必須の**		nonNegativeInteger
row	**必須の**		nonNegativeInteger
width	**必須の**		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>
```

親 要素

Layout 要素

License 要素

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>
```

親 要素

Extension 要素

子 要素

OptionCode 要素

ListValue 要素

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>
```

親 要素

Attribute 要素, ListValue 要素, Parameter 要素, Attribute 要素, ListValue 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

MapValue 要素

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>
</xs:sequence>
</xs:element>
```

親 要素

ListValue 要素, Attribute 要素, ListValue 要素, Parameter 要素, Attribute 要素, ListValue 要素

子 要素

MapEntry 要素

MapEntry 要素

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>
</xs:sequence>
</xs:element>
```

親 要素

MapValue 要素

子 要素

KeyValue 要素, StructuredValue 要素

KeyValue 要素

The key value in a map entry.

テーブル A-73
の属性 KeyValue

属性	使用	説明	有効値
value	**必須の**		string

XML 表示

```
<xs:element name="KeyValue" type="KEY-VALUE">
    <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

MapEntry 要素

StructuredValue 要素

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>
    </xs:sequence>
</xs:element>
```

親 要素

MapEntry 要素

子 要素

Attribute 要素

Attribute 要素

テーブル A-74
の属性 Attribute

属性	使用	説明	有効値
name	**必須の**		string
value	オプション		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>
```

親 要素

StructuredValue 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

ListValue 要素

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>
```

親 要素

Attribute 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

Menu 要素

テーブル A-75
の属性 Menu

属性	使用	説明	有効値
id	**必須の**		string
label	**必須の**		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string
offset	optional		nonNegativeInteger
separatorAfter	optional		boolean
separatorBefore	optional		boolean
showIcon	optional		boolean
showLabel	optional		boolean
systemMenu	**必須の**		**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>
```

親 要素

Controls 要素

MenuItem 要素

テーブル A-76
の属性 MenuItem

属性	使用	説明	有効値
action	**必須の**		string
customMenu	optional		string
offset	optional		nonNegativeInteger
separatorAfter	optional		boolean
separatorBefore	optional		boolean
showIcon	optional		boolean

属性	使用	説明	有効値
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.streamProper–ties**

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: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>
```

親 要素

Controls 要素

MissingValues 要素

テーブル A-77
の属性 MissingValues

属性	使用	説明	有効値
treatNullAsMissing	オプション		boolean
treatWhitespaceAsMissing	オプション		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:attribute name="treatNullAsMissing" type="xs:boolean" default="true"></xs:attribute>
    <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean"
    default="false"></xs:attribute>
</xs:element>
```

親 要素

Field 要素

子 要素

Range 要素, Values 要素

Range 要素

テーブル A-78
の属性 Range

属性	使用	説明	有効値
maxValue	**必須の**		string
minValue	**必須の**		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>
```

親 要素

MissingValues 要素

Values 要素

テーブル A-79
の属性 Values

属性	使用	説明	有効値
count	オプション		nonNegativeInte-ger

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:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

親 要素

MissingValues 要素

子 要素

Value 要素

Value 要素

テーブル A-80
の属性 Value

属性	使用	説明	有効値
code	**必須の**		integer
displayLabel	オプション		string
flagValue	オプション		boolean
value	**必須の**		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>
```

親 要素

Values 要素

子 要素

DisplayLabel 要素

DisplayLabel 要素

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.

テーブル A-81
の属性 DisplayLabel

属性	使用	説明	有効値
lang	オプション		NMTOKEN
value	**必須の**		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>
```

親 要素

Value 要素

ModelBuilder 要素

テーブル A-82
の属性 ModelBuilder

属性	使用	説明	有効値
allowNoInputs	optional		boolean
allowNoOutputs	optional		boolean
miningFunctions	**必須の**		いずれでも可
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: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>
```

親 要素

Node 要素

子 要素

Algorithm 要素, AutoModeling 要素, ModelEvaluation 要素, ModelFields 要素, ModelGeneration 要素, ModelingFields 要素

Algorithm 要素

テーブル A-83
の属性 Algorithm

属性	使用	説明	有効値
label	**必須の**		string
labelKey	optional		string
value	**必須の**		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>
```

親 要素

ModelBuilder 要素

ModelingFields 要素

テーブル A-84
の属性 ModelingFields

属性	使用	説明	有効値
controlsId	**必須の**		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>
```

親 要素

ModelBuilder 要素

子 要素

InputFields 要素, OutputFields 要素

関連 要素

DocumentGeneration 要素, ModelGeneration 要素, ModelEvaluation 要素

InputFields 要素

テーブル A-85
の属性 InputFields

属性	使用	説明	有効値
label	**必須の**		string
labelKey	optional		string
multiple	**必須の**		boolean
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean

属性	使用	説明	有効値
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	**必須の**		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>
```

親 要素

ModelingFields 要素

関連 要素

OutputFields 要素

OutputFields 要素

テーブル A-86
の属性 OutputFields

属性	使用	説明	有効値
label	**必須の**		string
labelKey	optional		string
multiple	**必須の**		boolean
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	**必須の**		string

属性	使用	説明	有効値
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>
```

親 要素

ModelingFields 要素

関連 要素

InputFields 要素

ModelGeneration 要素

テーブル A-87
の属性 ModelGeneration

属性	使用	説明	有効値
controlsId	**必須の**		string

XML 表示

```
<xs:element name="ModelGeneration">
    <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

ModelBuilder 要素

関連 要素

DocumentGeneration 要素, ModelingFields 要素, ModelEvaluation 要素

ModelFields 要素

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>
```

親 要素

ModelBuilder 要素

子 要素

AddField 要素, ChangeField 要素, ForEach 要素, RemoveField 要素

ModelEvaluation 要素

テーブル A-88
の属性 ModelEvaluation

属性	使用	説明	有効値
container	**必須の**		いずれでも可
controlsId	**必須の**		string
outputContainer	optional		いずれでも可

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>
```

親 要素

ModelBuilder 要素

子 要素

AdjustedPropensity 要素, RawPropensity 要素, VariableImportance 要素

関連 要素

DocumentGeneration 要素, ModelingFields 要素, ModelGeneration 要素

RawPropensity 要素

テーブル A-89
の属性 RawPropensity

属性	使用	説明	有効値
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>
```

親 要素

ModelEvaluation 要素

AdjustedPropensity 要素

テーブル A-90
の属性 AdjustedPropensity

属性	使用	説明	有効値
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>
```

親 要素

ModelEvaluation 要素

VariableImportance 要素

テーブル A-91
の属性 VariableImportance

属性	使用	説明	有効値
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>
```

親 要素

ModelEvaluation 要素

AutoModeling 要素

テーブル A-92
の属性 AutoModeling

属性	使用	説明	有効値
enabledByDefault	optional		いずれでも可

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:element ref="Constraint" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="enabledByDefault" use="optional"></xs:attribute>
</xs:element>
```

親 要素

ModelBuilder 要素

子 要素

Constraint 要素, ExpertSettings 要素, PropertyMap 要素, SimpleSettings 要素

SimpleSettings 要素

XML 表示

```
<xs:element name="SimpleSettings">
  <xs:sequence>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

親 要素

AutoModeling 要素

子 要素

PropertyGroup 要素

ExpertSettings 要素

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>
```

親 要素

AutoModeling 要素

子 要素

Condition 要素, PropertyGroup 要素

PropertyMap 要素

XML 表示

```
<xs:element name="PropertyMap" minOccurs="0">
    <xs:sequence>
        <xs:element name="PropertyMapping" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

AutoModeling 要素

子 要素

PropertyMapping 要素

PropertyMapping 要素

テーブル A-93
の属性 PropertyMapping

属性	使用	説明	有効値
property	**必須の**		string
systemProperty	**必須の**		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>
```

親 要素

PropertyMap 要素

ModelOutput 要素

テーブル A-94
の属性 ModelOutput

属性	使用	説明	有効値
deprecatedScriptNames	optional		string
helpLink	optional		string

属性	使用	説明	有効値
id	**必須の**		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>
```

親 要素

Extension 要素

子 要素

Constructors 要素, Containers 要素, ModelProvider 要素, Properties 要素, UserInterface 要素

関連 要素

DocumentOutput 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, Node 要素

Containers 要素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
```

```
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Container"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

ModelOutput 要素

子 要素

Container 要素

ModelProvider 要素

テーブル A-95
の属性 ModelProvider

属性	使用	説明	有効値
container	**必須の**		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>
```

親 要素

DocumentOutput 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

ModelType 要素

Defines a new model type

テーブル A-96
の属性 ModelType

属性	使用	説明	有効値
format	**必須の**		**utf8** **binary**
id	**必須の**		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>
```

親 要素

ContainerTypes 要素

関連 要素

DocumentType 要素

ModelViewerPanel 要素

テーブル A-97
の属性 ModelViewerPanel

属性	使用	説明	有効値
container	**必須の**		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>
```

親 要素

Tab 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

Module 要素

テーブル A-98
の属性 Module

属性	使用	説明	有効値
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>
```

親 要素

Execution 要素

子 要素

InputFiles 要素, OutputFiles 要素, StatusCodes 要素

MultiFieldChooserControl 要素

テーブル A-99
の属性 MultiFieldChooserControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

MultiItemChooserControl 要素

テーブル A-100
の属性 MultiItemChooserControl

属性	使用	説明	有効値
catalog	**必須の**		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	**必須の**		string
showLabel	optional		boolean

XML 表示

```
<xs:element name="MultiItemChooserControl">
    <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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

SingleItemChooserControl 要素

Node 要素

テーブル A-101
の属性 Node

属性	使用	説明	有効値
deprecatedScriptNames	optional		string
description	optional		string
descriptionKey	optional		string
helpLink	optional		string
id	**必須の**		string
label	**必須の**		string
labelKey	optional		string

属性	使用	説明	有効値
palette	optional		**import** **fieldOp** **recordOp** **modeling** **dbModeling** **graph** **output** **export** **modeling.classifica-tion** **modeling.association** **modeling.segmenta-tion** **modeling.auto**
relativePosition	optional		string
relativeTo	optional		string
scriptName	optional		string
type	**必須の**		**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>
```

親 要素

Extension 要素

子 要素

Constructors 要素, Containers 要素, DocumentBuilder 要素, Execution 要素, ModelBuilder 要素, ModelProvider 要素, OutputDataModel 要素, Properties 要素, UserInterface 要素

関連 要素

DocumentOutput 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素

Containers 要素

XML 表示

```
<xs:element name="Containers" minOccurs="0">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Container"></xs:element>
```

```
    </xs:sequence>
</xs:element>
```

親 要素

Node 要素

子 要素

Container 要素

Not 要素

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>
```

親 要素

And 要素, Command 要素, Constraint 要素, CreateDocument 要素, CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModel 要素, CreateModelApplier 要素, CreateModelOutput 要素, Enabled 要素, Option 要素, Or 要素, Run 要素, Visible 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

NumberFormat 要素

Defines format information for a numeric field.

テーブル A-102
の属性 NumberFormat

属性	使用	説明	有効値
decimalPlaces	**必須の**		nonNegativeInteger
decimalSymbol	**必須の**		**period** **comma**
formatType	**必須の**		**standard** **scientific** **currency**
groupingSymbol	**必須の**		**none** **period** **comma** **space**
name	**必須の**		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 要素

テーブル A-103
の属性 NumericInfo

属性	使用	説明	有効値
mean	オプション		double
standardDeviation	オプション		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>
```

親 要素

AddField 要素, ChangeField 要素, Field 要素

Option 要素

テーブル A-104
の属性 Option

属性	使用	説明	有効値
ifProperty	optional		string
unlessProperty	optional		string
value	**必須の**		

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>
```

親 要素

Run 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

OptionCode 要素

テーブル A-105
の属性 OptionCode

属性	使用	説明	有効値
code	オプション		long
description	オプション		string
type	オプション		**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>
```

親 要素

License 要素

Or 要素

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>
```

親 要素

And 要素, Command 要素, Constraint 要素, CreateDocument 要素, CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModel 要素, CreateModelApplier 要素, CreateModelOutput 要素, Enabled 要素, Not 要素, Option 要素, Run 要素, Visible 要素

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

OutputDataModel 要素

テーブル A-106
の属性 OutputDataModel

属性	使用	説明	有効値
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>
```

親 要素

Node 要素

OutputFiles 要素

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>
```

親 要素

Execution 要素, Module 要素

子 要素

ContainerFile 要素, DataFile 要素

Palette 要素

テーブル A-107
の属性 Palette

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string
id	**必須の**		string
label	**必須の**		string
labelKey	optional		string
position	optional		**atStart** **atEnd** **before** **after**
systemPalette	optional		**import** **fieldOp** **recordOp** **modeling** **dbModeling** **graph** **output** **export** **modeling.classifica-tion** **modeling.association** **modeling.segmenta-tion** **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 要素

Icon 要素

Parameters 要素

Configuration parameters from the extension node.

テーブル A-108
の属性 Parameters

属性	使用	説明	有効値
count	オプション		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 要素

Parameter 要素

Parameter 要素

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.

テーブル A-109
の属性 Parameter

属性	使用	説明	有効値
name	**必須の**		string
value	オプション		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>

親 要素

Parameters 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

PasswordBoxControl 要素

テーブル A-110
の属性 PasswordBoxControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

Properties 要素

XML 表示

```
<xs:element name="Properties">
    <xs:sequence>
        <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

DocumentOutput 要素, Execution 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

子 要素

Property 要素

PropertiesPanel 要素

テーブル A-111
の属性 PropertiesPanel

属性	使用	説明	有効値
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: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>
```

親 要素

PropertiesSubPanel 要素, Tab 要素

子 要素

ActionButton 要素, CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, ComboBoxControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, Enabled 要素, ExtensionObjectPanel 要素, Layout 要素, MultiFieldChooserControl 要素, MultiItemChooserControl 要素, PasswordBoxControl 要素, PropertiesPanel 要素, PropertiesSubPanel 要素, PropertyControl 要素, RadioButtonGroupControl 要素, SelectorPanel 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SingleItemChooserControl 要素, SpinnerControl 要素, StaticText 要素, SystemControls 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素, Visible 要素

関連 要素

PropertiesSubPanel 要素

PropertiesSubPanel 要素

テーブル A-112
の属性 PropertiesSubPanel

属性	使用	説明	有効値
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: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>
```

親 要素

PropertiesPanel 要素

子 要素

ActionButton 要素, CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, ComboBoxControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, Enabled 要素, ExtensionObjectPanel 要素, Layout 要素, MultiFieldChooserControl 要素, MultiItemChooserControl 要素, PasswordBoxControl 要素, PropertiesPanel 要素, PropertiesSubPanel 要素, PropertyControl 要素, RadioButtonGroupControl 要素, SelectorPanel 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SingleItemChooserControl 要素, SpinnerControl 要素, StaticText 要素, SystemControls 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素, Visible 要素

関連 要素

PropertiesPanel 要素

Property 要素

テーブル A-113
の属性 Property

属性	使用	説明	有効値
defaultValue	optional		
defaultValueKey	optional		string
deprecatedScriptNames	optional		string
description	optional		string
descriptionKey	optional		string
isList	optional		boolean
label	optional		string
labelKey	optional		string
max	optional		string
min	optional		string
name	**必須の**		string
scriptName	optional		string

属性	使用	説明	有効値
type	optional		string
valueType	オプション		**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>
```

親 要素

Properties 要素, PropertySets 要素

子 要素

DefaultValue 要素

関連 要素

PropertyType 要素

PropertyControl 要素

テーブル A-114
の属性 PropertyControl

属性	使用	説明	有効値
controlClass	**必須の**		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	**必須の**		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:attribute name="controlClass" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

PropertyGroup 要素

テーブル A-115
の属性 PropertyGroup

属性	使用	説明	有効値
label	optional		string
labelKey	optional		string
properties	**必須の**		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>
```

親 要素

SimpleSettings 要素, ExpertSettings 要素

PropertySets 要素

XML 表示

```
<xs:element name="PropertySets">
    <xs:sequence>
        <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"></xs:element>
```

```
    </xs:sequence>
</xs:element>
```

親 要素

CommonObjects 要素

子 要素

Property 要素

PropertyType 要素

テーブル A-116
の属性 PropertyType

属性	使用	説明	有効値
id	**必須の**		string
isKeyed	optional		boolean
isList	optional		boolean
max	optional		string
min	optional		string
valueType	オプション		**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>
```

親 要素

PropertyTypes 要素

子 要素

DefaultValue 要素, Enumeration 要素, Structure 要素

関連 要素

Property 要素

PropertyTypes 要素

XML 表示

```
<xs:element name="PropertyTypes">
    <xs:sequence>
        <xs:element ref="PropertyType" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

CommonObjects 要素

子 要素

PropertyType 要素

RadioButtonGroupControl 要素

テーブル A-117
の属性 RadioButtonGroupControl

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string

属性	使用	説明	有効値
falseLabel	optional		string
falseLabelKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
layoutByRow	optional		boolean
mnemonic	optional		string
mnemonicKey	optional		string
property	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

Range 要素

テーブル A-118
の属性 Range

属性	使用	説明	有効値
max	オプション		string
min	オプション		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>
```

親 要素

AddField 要素, MissingValues 要素, ChangeField 要素, MissingValues 要素, Field 要素, MissingValues 要素

Range 要素

テーブル A-119
の属性 Range

属性	使用	説明	有効値
maxValue	**必須の**		string
minValue	**必須の**		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>
```

親 要素

Field 要素

RemoveField 要素

テーブル A-120
の属性 RemoveField

属性	使用	説明	有効値
fieldRef	**必須の**		

XML 表示

```
<xs:element name="RemoveField">
    <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

親 要素

ForEach 要素, ModelFields 要素

Resources 要素

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>
```

親 要素

Extension 要素

子 要素

Bundle 要素, HelpInfo 要素, JarFile 要素, SharedLibrary 要素

Bundle 要素

テーブル A-121
の属性 Bundle

属性	使用	説明	有効値
id	**必須の**		string
path	**必須の**		
type	**必須の**		**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>
```

親 要素

Resources 要素

JarFile 要素

テーブル A-122
の属性 JarFile

属性	使用	説明	有効値
id	**必須の**		string
path	**必須の**		

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>
```

親 要素

Resources 要素

SharedLibrary 要素

テーブル A-123
の属性 SharedLibrary

属性	使用	説明	有効値
id	**必須の**		string
path	**必須の**		

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>
```

親 要素

Resources 要素

HelpInfo 要素

テーブル A-124
の属性 HelpInfo

属性	使用	説明	有効値
default	optional		string
helpset	optional		
id	optional		string
missing	optional		string

属性	使用	説明	有効値
path	optional		
type	**必須の**		**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>
```

親 要素

Resources 要素

Run 要素

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>
```

親 要素

Executable 要素

子 要素

And 要素, Command 要素, Condition 要素, Not 要素, Option 要素, Or 要素, StatusCodes 要素

SelectorPanel 要素

テーブル A-125
の属性 SelectorPanel

属性	使用	説明	有効値
control	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Selector 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

Selector 要素

テーブル A-126
の属性 Selector

属性	使用	説明	有効値
controlValue	**必須の**		string
panelId	**必須の**		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>
```

親 要素

SelectorPanel 要素

ServerDirectoryChooserControl 要素

テーブル A-127
の属性 ServerDirectoryChooserControl

属性	使用	説明	有効値
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	**必須の**		**open** **save** **import** **export**
property	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

ServerFileChooserControl 要素

テーブル A-128
の属性 ServerFileChooserControl

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean

属性	使用	説明	有効値
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
mode	**必須の**		**open** **save** **import** **export**
property	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

SetContainer 要素

テーブル A-129
の属性 SetContainer

属性	使用	説明	有効値
source	**必須の**		string
target	**必須の**		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>
```

親 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

関連 要素

SetProperty 要素

SetProperty 要素

テーブル A-130
の属性 SetProperty

属性	使用	説明	有効値
source	**必須の**		string
target	**必須の**		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>
```

親 要素

CreateDocumentOutput 要素, CreateInteractiveDocumentBuilder 要素, CreateInteractiveModelBuilder 要素, CreateModelApplier 要素, CreateModelOutput 要素

関連 要素

SetContainer 要素

SingleFieldChooserControl 要素

テーブル A-131
の属性 SingleFieldChooserControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

SingleFieldValueChooserControl 要素

テーブル A-132
の属性 SingleFieldValueChooserControl

属性	使用	説明	有効値
description	optional		string
descriptionKey	optional		string

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

SingleItemChooserControl 要素

テーブル A-133
の属性 SingleItemChooserControl

属性	使用	説明	有効値
catalog	**必須の**		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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

MultiItemChooserControl 要素

SpinnerControl 要素

テーブル A-134
の属性 SpinnerControl

属性	使用	説明	有効値
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	**必須の**		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: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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素

SPSSDataFormat 要素

XML 表示

```
<xs:element name="SPSSDataFormat"></xs:element>
```

親 要素

DataFormat 要素

StaticText 要素

テーブル A-135
の属性 StaticText

属性	使用	説明	有効値
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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, SystemControls 要素, TabbedPanel 要素, TextBrowserPanel 要素

StatusCode 要素

テーブル A-136
の属性 StatusCode

属性	使用	説明	有効値
code	**必須の**		integer
message	optional		string

属性	使用	説明	有効値
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>
```

親 要素

StatusCodes 要素

StatusCodes 要素

テーブル A-137
の属性 StatusCodes

属性	使用	説明	有効値
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>
```

親 要素

Module 要素, Run 要素

子 要素

StatusCode 要素

StatusDetail 要素

Supplementary information about a progress or other conditions.

テーブル A-138
の属性 StatusDetail

属性	使用	説明	有効値
destination	オプション		**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 要素

Diagnostic 要素

Diagnostic 要素

テーブル A-139
の属性 Diagnostic

属性	使用	説明	有効値
code	**必須の**		integer
severity	オプション		**unknown** **information** **warning** **error** **fatal**
source	オプション		string
subCode	オプション		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>
```

親 要素

StatusDetail 要素

子 要素

Message 要素, Parameter 要素

Message 要素

テーブル A-140
の属性 Message

属性	使用	説明	有効値
lang	オプション		NMTOKEN

XML 表示

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
    <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>
```

親 要素

Diagnostic 要素

Parameter 要素

XML 表示

```
<xs:element name="Parameter" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>
```

親 要素

Diagnostic 要素

Structure 要素

XML 表示

```
<xs:element name="Structure">
    <xs:sequence>
        <xs:element ref="Attribute" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

PropertyType 要素

子 要素

Attribute 要素

StructuredValue 要素

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>
        </xs:sequence>
    </xs:element>
```

親 要素

ListValue 要素, Attribute 要素, ListValue 要素, Parameter 要素, Attribute 要素, ListValue 要素

子 要素

Attribute 要素

Attribute 要素

テーブル A-141 の属性 Attribute

属性	使用	説明	有効値
name	**必須の**		string
value	オプション		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>
```

親 要素

StructuredValue 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

ListValue 要素

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>
```

親 要素

Attribute 要素

子 要素

DatabaseConnectionValue 要素, ListValue 要素, MapValue 要素, StructuredValue 要素, Value 要素

SystemControls 要素

テーブル A-142
の属性 SystemControls

属性	使用	説明	有効値
controlsId	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, TabbedPanel 要素, TextBrowserPanel 要素

Tab 要素

テーブル A-143
の属性 Tab

属性	使用	説明	有効値
helpLink	optional		string
id	optional		string
label	**必須の**		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>
```

親 要素

Tabs 要素

子 要素

ExtensionObjectPanel 要素, ModelViewerPanel 要素, PropertiesPanel 要素, TabbedPanel 要素, TextBrowserPanel 要素

TabbedPanel 要素

テーブル A-144
の属性 TabbedPanel

属性	使用	説明	有効値
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>
```

親 要素

Tab 要素

子 要素

Enabled 要素, Layout 要素, Tabs 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TextBrowserPanel 要素

TableControl 要素

テーブル A-145
の属性 TableControl

属性	使用	説明	有効値
columns	optional		positiveInteger
columnWidths	optional		string
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string

属性	使用	説明	有効値
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要

素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TextAreaControl 要素, TextBoxControl 要素

Tabs 要素

テーブル A-146
の属性 Tabs

属性	使用	説明	有効値
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>
```

親 要素

TabbedPanel 要素, UserInterface 要素

子 要素

Tab 要素

TextAreaControl 要素

テーブル A-147
の属性 TextAreaControl

属性	使用	説明	有効値
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	**必須の**		string

属性	使用	説明	有効値
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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素,

SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextBoxControl 要素

TextBoxControl 要素

テーブル A-148
の属性 TextBoxControl

属性	使用	説明	有効値
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	**必須の**		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>
```

親 要素

PropertiesPanel 要素, PropertiesSubPanel 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, MultiFieldChooserControl 要素, PasswordBoxControl 要素, PropertyControl 要素, RadioButtonGroupControl 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SpinnerControl 要素, TableControl 要素, TextAreaControl 要素

TextBrowserPanel 要素

テーブル A-149
の属性 TextBrowserPanel

属性	使用	説明	有効値
columns	optional		string
container	**必須の**		string
rows	optional		string
textFormat	**必須の**		**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>
```

親 要素

Tab 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

関連 要素

ActionButton 要素, ComboBoxControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, SelectorPanel 要素, StaticText 要素, SystemControls 要素, TabbedPanel 要素

ToolbarItem 要素

テーブル A-150
の属性 ToolbarItem

属性	使用	説明	有効値
action	**必須の**		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>
```

親 要素

Controls 要素

UserInterface 要素

テーブル A-151
の属性 UserInterface

属性	使用	説明	有効値
actionHandler	optional		いずれでも可
frameClass	optional		いずれでも可

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>
```

親 要素

DocumentOutput 要素, Extension 要素, InteractiveDocumentBuilder 要素, InteractiveModelBuilder 要素, ModelOutput 要素, Node 要素

子 要素

Controls 要素, Icons 要素, Tabs 要素

UTF8Format 要素

XML 表示

```
<xs:element name="UTF8Format"></xs:element>
```

親 要素

FileFormatType 要素

Value 要素

A simple value.

テーブル A-152
の属性 Value

属性	使用	説明	有効値
value	**必須の**		string

XML 表示

```
<xs:element name="Value" type="SIMPLE-VALUE">
    <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

親 要素

ListValue 要素, Attribute 要素, ListValue 要素, Parameter 要素, Attribute 要素, ListValue 要素

Values 要素

XML 表示

```
<xs:element name="Values">
    <xs:sequence>
        <xs:element name="Value" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
</xs:element>
```

親 要素

AddField 要素, MissingValues 要素, ChangeField 要素, MissingValues 要素, Field 要素, MissingValues 要素

子 要素

Value 要素

Value 要素

テーブル A-153
の属性 Value

属性	使用	説明	有効値
flagProperty	オプション		**trueValue** **falseValue**
value	**必須の**		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>
```

親 要素

Values 要素

Values 要素

テーブル A-154
の属性 Values

属性	使用	説明	有効値
code	**必須の**		integer
displayLabel	オプション		string
flagValue	オプション		boolean
value	**必須の**		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>
```

親 要素

Field 要素

子 要素

DisplayLabel 要素

DisplayLabel 要素

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.

テーブル A-155
の属性 DisplayLabel

属性	使用	説明	有効値
lang	オプション		NMTOKEN
value	**必須の**		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>
```

親 要素

Values 要素

Visible 要素

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>
```

親 要素

ActionButton 要素, CheckBoxControl 要素, CheckBoxGroupControl 要素, ClientDirectoryChooserControl 要素, ClientFileChooserControl 要素, ComboBoxControl 要素, DBConnectionChooserControl 要素, DBTableChooserControl 要素, ExtensionObjectPanel 要素, ModelViewerPanel 要素, MultiFieldChooserControl 要素, MultiItemChooserControl 要素, PasswordBoxControl 要素, PropertiesPanel 要素, PropertiesSubPanel 要素, PropertyControl 要素, RadioButtonGroupControl 要素, SelectorPanel 要素, ServerDirectoryChooserControl 要素, ServerFileChooserControl 要素, SingleFieldChooserControl 要素, SingleFieldValueChooserControl 要素, SingleItemChooserControl 要素, SpinnerControl 要素, StaticText

要素, SystemControls 要素, TabbedPanel 要素, TableControl 要素, TextAreaControl 要素, TextBoxControl 要素, TextBrowserPanel 要素, ItemChooserControl タイプ

子 要素

And 要素, Condition 要素, Not 要素, Or 要素

拡張 タイプ

拡張タイプは、属性および子要素を追加することによって、XML 文書の要素を拡張できます。 XML 文書で拡張タイプを使用するには、その要素の属性が xsi:type である拡張タイプを指定します。 その拡張タイプで定義された属性および要素を使用できるようになります。

ItemChooserControl タイプ

テーブル A-156
の属性 ItemChooserControl

属性	使用	説明	有効値
catalog	**必須の**		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	**必須の**		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>
```

拡張

ComboBoxControl 要素

子 要素

Enabled 要素, Layout 要素, Visible 要素

付録

B

注意事項

この情報は、世界各国で提供される製品およびサービス向けに作成されています。

IBM はこのドキュメントで説明する製品、サービス、機能は他の国では提供していない場合があります。現在お住まいの地域で利用可能な製品、サービス、および、情報については、お近くの IBM の担当者にお問い合わせください。IBM 製品、プログラム、またはサービスに対する参照は、IBM 製品、プログラム、またはサービスのみが使用することができることを説明したり意味するものではありません。IBM の知的所有権を侵害しない機能的に同等の製品、プログラム、またはサービスを代わりに使用することができます。ただし、IBM 以外の製品、プログラム、またはサービスの動作を評価および確認するのはユーザーの責任によるものです。

IBMは、本ドキュメントに記載されている内容に関し、特許または特許出願中の可能性があります。本ドキュメントの提供によって、これらの特許に関するいかなる権利も使用者に付与するものではありません。ライセンスのお問い合わせは、書面にて、下記住所に送ることができます。

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

2 バイト文字セット (DBCS) 情報についてのライセンスに関するお問い合わせは、お住まいの国の IBM Intellectual Property Department に連絡するか、書面にて下記宛先にお送りください。

神奈川県大和市下鶴間1623番14号 日本アイ・ビー・エム株式会社 法務・知的財産 知的財産権ライセンス渉外

以下の条項はは、イギリスまたはこのような条項が法律に反する他の国では適用されません。 International Business Machines は、明示的または黙示的に関わらず、第三者の権利の侵害しない、商品性または特定の目的に対する適合性の暗黙の保証を含むがこれに限定されない、いかなる保証なく、本出版物を「そのまま」提供します一部の州では、特定の取引の明示的または暗示的な保証の免責を許可していないため、この文が適用されない場合があります。

この情報には、技術的に不適切な記述や誤植を含む場合があります。情報については変更が定期的に行われます。これらの変更は本書の新版に追加されます。IBM は、本書に記載されている製品およびプログラムについて、事前の告知なくいつでも改善および変更を行う場合があります。

© Copyright IBM Corporation 1994, 2012.

IBM 以外の Web サイトに対するこの情報内のすべての参照は、便宜上提供されているものであり、決してそれらの Web サイトを推奨するものではありません。これらの Web サイトの資料はこの IBM 製品の資料に含まれるものではなく、これらの Web サイトの使用はお客様の責任によるものとします。

IBM はお客様に対する一切の義務を負うことなく、自ら適切と考える方法で、情報を使用または配布することができるものとします。

本プログラムのライセンス取得者が (i) 別途作成されたプログラムと他のプログラム (本プログラムを含む) との間の情報交換および (ii) 交換された情報の相互利用を目的とした本プログラムに関する情報の所有を希望する場合、下記住所にお問い合わせください。

IBM Software Group, Attention:Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

上記のような情報は、該当する条項および条件に従い、有料で利用できるものとします。

本ドキュメントに記載されている許可されたプログラムおよびそのプログラムに使用できるすべてのライセンス認証された資料は、IBM Customer Agreement、IBM International Program License Agreement、および当社とかわした同等の契約の条件に基づき、IBM によって提供されます。

ここに記載されているパフォーマンスデータは、すべて管理環境下で確認されたものです。そのため、他の操作環境で得られた結果は大きく異なる可能性があります。開発レベルのシステムで測定が行われている場合があり、これらの測定値は一般に利用可能なシステムと同じであることを保証するものではありません。また、測定値が推定値である可能性があり、実際の結果は異なる場合があります。本ドキュメントのユーザーは、特定の環境に適したデータを検証する必要があります。

IBM 以外の製品に関する情報は、それらの製品の供給業者、公開済みの発表、または公開で使用できるソースから取得しています。IBM は、それらの製品のテストは行っておらず、IBM 以外の製品に関連する性能、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給業者に通知する必要があります。

IBM の将来の方向性または意向に関する記述については、予告なく変更または取り消すことがあり、目的や目標のみを示すものです。

この情報には、日常の業務処理で用いられるデータや報告書の例が含まれています。できる限り詳細に説明するため、例には、個人、企業、ブランド、製品などの名前が使用されています。これらの名称はすべて架空のものであり、実際の企業で使用される名称および住所とは一切関係ありません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーのイラストが表示されない場合があります。

商標

IBM、IBM ロゴ、および ibm.com、SPSS は、世界の多くの国で登録された IBM Corporation の商標です。IBM の商標の現在のリストは、http://www.ibm.com/legal/copytrade.shtml を参照してください。

Intel、Intel のロゴ、Intel Inside、Intel Inside のロゴ、Intel Centrino、Intel Centrino のロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、米国およびその他の国の Intel Corporation または関連会社の商標または登録商標です。

Linux は、米国およびその他の国における Linus Torvalds の登録商標です。

Microsoft、Windows、Windows NT、および Windows のロゴは、米国およびその他の国における Microsoft 社の商標です。

UNIX は、米国およびその他の国における The Open Group の登録商標です。

Java およびすべての Java ベースの商標およびロゴは、米国およびその他の国の Sun Microsystems, Inc. の商標です。

その他の製品名およびサービス名等は、IBM または他の会社の商標です。

索引

下位互換性、保持, 105
評価文字列, 88
複合条件, 103
選択情報, 182
報告書, 54
文字列
  暗号化, 83
  評価, 88
削除
  パレットおよびサブパレット, 64
商標, 445
拡張, 1
  配布, 278
  アンインストール, 279
  インストール, 279
  オブジェクト パネル, 162
  下位互換性の保持, 105
  ローカライズ, 230
操作
  ボタン, 168
構築
  インタラクティブ モデル, 109, 120
  モデル, 109
言語
  設定, 230
  コード、ISO 規格, 232
傾向 (調整なし), 89
実行、外部 (実行プロセス), 275
[注釈] タブ、ノードのダイアログ ボックス, 30
[出力] タブ、マネージャ領域, 136

Action 要素, 280
Action 要素、特性ファイル, 55
ActionButton 要素, 281
ActionButton 要素、特性ファイル, 169
Actions 要素, 282
Actions 要素、特性ファイル, 55
AddField 要素, 282
AddField 要素、特性ファイル, 89, 95
AdjustedPropensity 要素, 372
Algorithm 要素, 367
Algorithm 要素、設定ファイル, 111
And 要素, 287
And 要素、特性ファイル, 98
反復関数、API, 247
進行関数、API, 247
Attribute 要素, 287, 358, 426
Attribute 要素 (カタログ)、特性ファイル, 56
Attribute 要素、特性ファイル, 85
AutoModeling 要素, 373
Automodeling 要素、設定ファイル, 125
BinaryFormat 要素, 288
Bundle 要素, 407
Bundle 要素、特性ファイル, 47
C ベース API, 5
C++
  language, 242
  ヘルパー, 266
Catalog 要素, 288
Catalog 要素、特性ファイル, 56
Catalogs 要素, 289
Catalogs 要素、特性ファイル, 56
Cell 要素, 353
Cell 要素、特性ファイル, 209
ChangeField 要素, 289
ChangeField 要素、特性ファイル, 91
CheckBoxControl 要素, 293
CheckBoxControl 要素、特性ファイル, 178
CheckBoxGroupControl 要素, 294
CheckBoxGroupControl 要素、特性ファイル, 179
ClientDirectoryChooserControl 要素, 296
ClientDirectoryChooserControl 要素、特性ファイル, 180
ClientFileChooserControl 要素, 297
ClientFileChooserControl 要素、特性ファイル, 181
ColumnControl 要素、特性ファイル, 201
ComboBoxControl 要素, 298
ComboBoxControl 要素、特性ファイル, 182
Command 要素, 300
CommonObjects 要素, 300
CommonObjects 要素、特性ファイル, 49
Condition 要素, 301
Condition 要素、特性ファイル, 98
Constraint 要素, 303
Constraint 要素、設定ファイル, 132
Constructors 要素, 304
Constructors 要素、設定ファイル, 137
container
  内容、調査, 276
  タイプ, 53
  ファイル, 78
Container 要素, 305
Container 要素、特性ファイル, 73
ContainerFile 要素, 305
Containers 要素, 329, 350-351, 376, 384
Containers 要素、特性ファイル, 73
ContainerTypes 要素, 306
ContainerTypes 要素、特性ファイル, 53
Controls 要素, 306
Controls 要素、特性ファイル, 149

CreateDocument 要素, 307
CreateDocument 要素、設定ファイル, 137
CreateDocumentOutput 要素, 307
CreateDocumentOutput 要素、設定ファイル, 138
CreateInteractiveDocumentBuilder 要素, 308
CreateInteractiveModelBuilder 要素, 309
CreateInteractiveModelBuilder 要素、設定ファイル, 122
CreateModel 要素, 310
CreateModel 要素、設定ファイル, 137
CreateModelApplier 要素, 311
CreateModelApplier 要素、設定ファイル, 140
CreateModelOutput 要素, 312
CreateModelOutput 要素、設定ファイル, 137
data
  タイプ, 251
  変換ノード, 12, 37, 67
  ライター ノード, 15, 67
  リーダー ノード, 12, 36, 67
DatabaseConnectionValue 要素, 313
DataFile 要素, 314
DataFormat 要素, 314
DataModel 要素, 315
DBConnectionChooserControl 要素, 322
DBConnectionChooserControl 要素、特性ファイル, 183
DBTableChooserControl 要素, 323
DBTableChooserControl 要素、特性ファイル, 185
DefaultValue 要素, 324
DefaultValue 要素、特性ファイル, 77
DelimitedDataFormat 要素, 326
Diagnostic 要素, 333, 424
Diagnostic 要素、状況詳細ドキュメント, 265
DisplayLabel 要素, 327, 365, 439
DocumentBuilder 要素, 327
DocumentBuilder 要素、設定ファイル, 135
DocumentGeneration 要素, 328
DocumentGeneration 要素、設定ファイル, 135
DocumentOutput 要素, 328
DocumentOutput 要素、設定ファイル, 136
DocumentType 要素, 330
DocumentType 要素、特性ファイル, 54
Enabled 要素, 330
Enabled 要素、特性ファイル, 217
Enum 要素, 332
Enum 要素、特性ファイル, 84
Enumeration 要素, 331
Enumeration 要素、特性ファイル, 84
ErrorDetail 要素, 332
Exclude 要素、特性ファイル, 96
Executable 要素, 334
Execution 要素, 335
Execution 要素、特性ファイル, 75
ExpertSettings 要素, 374
ExpertSettings 要素、設定ファイル, 129
extension
  フォルダ, 6
  モジュール, 242
Extension 要素, 335
Extension 要素、特性ファイル, 45
extension.xml file, 7, 42
ExtensionDetails 要素, 336
ExtensionDetails 要素、特性ファイル, 45
ExtensionObjectPanel 要素, 337
ExtensionObjectPanel 要素、特性ファイル, 163
Field 要素, 320, 337
FieldFormats 要素, 316, 341
FieldGroup 要素, 318, 342, 344
FieldGroups 要素, 317, 343
FieldName 要素, 319, 343, 345
Fields 要素, 319
FieldSet 要素、特性ファイル, 96
FileFormatType 要素, 345
FileFormatTypes 要素, 346
ForEach 要素, 346
ForEach 要素、特性ファイル, 92, 95
HelpInfo 要素, 408
HelpInfo の要素、設定ファイル, 226
HTML Help
  リンク先, 224
  ローカライズ, 236
Icon 要素, 347
Icons 要素, 348
Icons 要素、特性ファイル, 148
Identifier 要素, 325
Include 要素、特性ファイル, 96
InputFields 要素, 368
InputFields 要素、設定ファイル, 113
InputFiles 要素, 348
InputFiles 要素、特性ファイル, 77
interactive
  モデル、構築, 109, 120
InteractiveDocumentBuilder 要素, 349
InteractiveModelBuilder 要素, 350
InteractiveModelBuilder 要素、設定ファイル, 123
ISO 規格、言語コード, 232
ItemChooserControl タイプ, 441
JarFile 要素, 408
JarFile 要素、特性ファイル, 48
Java, 7
  API, 5
  クラス, 47-48, 55, 81, 146, 162, 191, 223

JavaHelp
  リンク先, 225
  ローカライズ, 237
KeyValue 要素, 357
Layout 要素, 352
Layout 要素、特性ファイル, 208
License 要素, 354
ListValue 要素, 354, 359, 427
MapEntry 要素, 356
MapValue 要素, 355
menu
  項目、カスタム, 16, 151
  領域、ダイアログ ボックス, 28
Menu 要素, 360
Menu 要素、特性ファイル, 150
MenuItem 要素, 361
MenuItem 要素、特性ファイル, 151
Message 要素, 334, 424
Message 要素、状況詳細ドキュメント, 265
MissingValues 要素, 285, 291, 339, 363
model
  署名, 116
  アプライヤ ノード, 15, 67, 107-108, 140
  出力オブジェクト, 107
  タイプ, 54
  ナゲット, 12
  ビルダー ノード, 12, 38, 67, 107-109
ModelBuilder 要素, 366
ModelBuilder 要素、設定ファイル, 109
ModelDetail 要素, 311
ModelEvaluation 要素, 371
ModelField 要素, 286, 292, 340
ModelFields 要素, 371
ModelFields 要素、設定ファイル, 116
ModelGeneration 要素, 370
ModelGeneration 要素、設定ファイル, 116
ModelingFields 要素, 368
ModelingFields 要素、設定ファイル, 112
ModelOutput 要素, 375
ModelOutput 要素、設定ファイル, 119
ModelProvider 要素, 377
ModelProvider 要素、特性ファイル, 71
ModelType 要素, 377
ModelType 要素、特性ファイル, 54
ModelViewerPanel 要素, 378
ModelViewerPanel 要素、特性ファイル, 167
Module 要素, 379
Module 要素、特性ファイル, 78
MultiFieldChooserControl 要素, 380
MultiFieldChooserControl
 要素、特性ファイル, 186
MultiItemChooserControl 要素, 381
MultiItemChooserControl
 要素、特性ファイル, 189
node
  attributes, 67
Node 要素, 382
Node 要素、特性ファイル, 66
nodes, 4, 10
  定義, 66
  CLEF 拡張のテスト, 274
  アンサンブル, 125
  データ変換, 12
  データ ライター, 15
  データ リーダー, 12
  ドキュメント ビルダー, 14
  モデル アプライヤ, 15
  モデル ビルダー, 12
Not 要素, 385
Not 要素、特性ファイル, 98
NumberFormat 要素, 316, 341, 385
NumericInfo 要素, 386
Object Definition セクション、特性ファイル,
 65
Option 要素, 387
OptionCode 要素, 388
Or 要素, 388
Or 要素、特性ファイル, 98
output
  ドキュメント （XML）, 257
  ファイル, 5, 76
OutputDataModel 要素, 389
OutputDataModel 要素、特性ファイル, 81
OutputFields 要素, 369
OutputFields 要素、設定ファイル, 115
OutputFiles 要素, 390
OutputFiles 要素、特性ファイル, 78
Palette 要素, 390
Palette 要素、特性ファイル, 59
Palettes 要素、特性ファイル, 59
Parameter 要素, 334, 392, 425
Parameter 要素、状況詳細ドキュメント, 265
Parameters 要素, 391
PasswordBoxControl 要素, 393
PasswordBoxControl 要素、特性ファイル, 190
PMML 形式、モデル出力, 71, 166
Properties 要素, 394
Properties 要素、特性ファイル, 71
  ランタイム, 76
PropertiesPanel 要素, 395
PropertiesPanel 要素、特性ファイル
  タブ またはプロパティ サブパネルから使用,
   165
  ネスト, 175
PropertiesSubPanel 要素, 396
PropertiesSubPanel 要素、特性ファイル, 173
Property 要素, 398

Property 要素、特性ファイル, 71
  ランタイム, 76
PropertyControl 要素, 400
PropertyControl 要素、特性ファイル, 191
PropertyGroup 要素, 401
PropertyGroup 要素、設定ファイル, 129
PropertyMap 要素, 375
PropertyMapping 要素, 375
PropertySet 要素、特性ファイル, 52
PropertySets 要素, 401
PropertySets 要素、特性ファイル, 52
PropertyType 要素, 402
PropertyType 要素、特性ファイル, 50
PropertyTypes 要素, 403
PropertyTypes 要素、特性ファイル, 50
PSAPI, 5
RadioButtonGroupControl 要素, 403
RadioButtonGroupControl
 要素、特性ファイル, 192
Range 要素, 364, 405-406
RawPropensity 要素, 372
RemoveField 要素, 406
RemoveField 要素、特性ファイル, 92
Resources 要素, 406
Resources 要素、特性ファイル, 46
Run 要素, 409
Selector 要素, 411
SelectorPanel 要素, 410
ServerDirectoryChooserControl 要素, 411
ServerDirectoryChooserControl
 要素、特性ファイル, 194
ServerFileChooserControl 要素, 412
ServerFileChooserControl
 要素、特性ファイル, 195
ServerTempDir 要素, 325
ServerTempFile 要素, 325
SetContainer 要素, 414
SetProperty 要素, 414
SharedLibrary 要素, 408
SharedLibrary 要素、特性ファイル, 48
SimpleSettings 要素, 374
SimpleSettings 要素、設定ファイル, 129
SingleFieldChooserControl 要素, 415
SingleFieldChooserControl
 要素、特性ファイル, 196
SingleFieldValueChooserControl 要素, 416
SingleItemChooserControl 要素, 418
SingleItemChooserControl
 要素、特性ファイル, 198
SpinnerControl 要素, 419
SpinnerControl 要素、特性ファイル, 199
SPSSDataFormat 要素, 420
SQL 生成ドキュメント、XML 出力, 263
SQL プッシュ バック, 253
StaticText 要素, 421
StaticText 要素、特性ファイル, 170
StatusCode 要素, 421
StatusCode 要素、特性ファイル, 79, 260
StatusCodes 要素, 422
StatusCodes 要素、特性ファイル, 79
StatusDetail 要素, 423
Structure 要素, 425
Structure 要素、特性ファイル, 85
StructuredValue 要素, 357, 425
SystemControls 要素, 428
SystemControls 要素、特性ファイル, 171
Tab 要素, 429
Tab 要素、特性ファイル, 156
TabbedPanel 要素, 429
TableControl 要素, 430
TableControl 要素、特性ファイル, 200
Tabs 要素, 432
Tabs 要素、特性ファイル, 155
TextAreaControl 要素, 432
TextAreaControl 要素、特性ファイル, 202
TextBoxControl 要素, 434
TextBoxControl 要素、特性ファイル, 203
TextBrowserPanel 要素, 435
TextBrowserPanel 要素、特性ファイル, 160
ToolbarItem 要素, 436
ToolbarItem 要素、特性ファイル, 153
UI コンポーネント
  システム コントロール, 170
  静的テキスト, 169
  操作ボタン, 168
User Interface セクション、特性ファイル,
 145
  カスタム パレット, 59
UserInterface 要素, 437
UserInterface 要素、特性ファイル, 75
  カスタム パレット, 59
UTF8Format 要素, 437
Value 要素, 365, 437-438
Values 要素, 364, 438-439
VariableImportance 要素, 373
Visible 要素, 440
Visible 要素、特性ファイル, 222
解析、XML, 268
XML(X)
  API の解析, 268
  出力ドキュメント, 257
  宣言、特性ファイル, 45

アイコン
  グラフィックの必要条件, 21
  生成されたモデル, 19
  領域、ダイアログ ボックス, 27

タイプ, 149
デザイン, 18
画像の作成, 21
ノード, 19
枠線、アイコン, 19
背景、アイコン, 20
画像、アイコン用に作成, 21
アクション
ハンドラ, 146
アクセス機能, 230, 238
アクセス キー, 157
アプリケーション プログラミング インターフェイス (API)
C ベース, 5
Java ベース, 5
PSAPI, 5, 242
クライアント側, 5, 241
サーバー側, 5, 242
ドキュメンテーション, 240
[アルゴリズム設定] ダイアログ, 128-129, 132
アルゴリズム、モデル ビルダー ノードの指定, 111
アンサンブル モデル構築ノード, 125
アーキテクチャ
サーバー側 API, 242
システム, 1

出力ウィンドウ, 142
カスタム, 223
デザイン, 32
対話ウィンドウ, 120

エラー処理, 267
エラー メッセージ、ローカライズ, 265
エラー詳細ドキュメント、XML 出力, 260

オブジェクト識別子, 66
出力オブジェクト
model, 12
ドキュメント, 14

カスタム出力ウィンドウ, 223
カタログ, 56

キャッシュ ステータス、ノード, 19
キャッシュ、データ, 261
キー プロパティ, 51, 86
キーボード ショートカット, 157

クライアント
ディレクトリ設定, 180
ファイル設定, 181
クライアント側 API, 5, 241
使用, 241
クラス, 241
クライアント側のコンポーネント, 1
クラス, 7
クライアント側 API, 241
グラフ, 54
グラフィックの必要条件、アイコン, 21
グリフ, 18
グループ、フィールド, 117-118

コメント行、設定ファイル, 43
コンストラクタ, 108
コンストラクタ、使用, 137
コンテナ, 53, 73
の内容の調査, 276
コントローラ, 175
単一項目設定, 198
複数項目設定, 189
選択情報, 182
属性, 177
クライアント ディレクトリ設定, 180
クライアント ファイル設定, 181
サーバー ディレクトリ設定, 194
サーバー ファイル設定, 195
スピン, 199
チェック ボックス, 178
チェック ボックス グループ, 179
テキスト領域, 202
テキスト ボックス, 203
データベース接続設定, 183
データベース テーブル設定, 185
テーブル, 200
パスワード ボックス, 190
単一フィールド設定, 196
複数フィールド設定, 186
プロパティ コントロール, 191
ラジオ ボタン グループ, 192
属性、コントローラ, 177
単一項目設定コントロール, 198
複数項目設定コントロール, 189
コントロールの順序、変更, 207
コントロール、ノードのダイアログ ボックス, 23
コントロール、画面プロパティ, 167
UI コンポーネント, 168
コントローラ, 175
プロパティ パネル, 172
画面コンポーネントの表示、制御, 222
コールバック関数、API, 242, 245

サブパレット
  非表示, 64
  削除, 64
  ノードの指定, 17, 59, 67
暗号化された文字列, 83
生成されたオブジェクト
  model, 109
  グラフまたはレポート, 135
自動化されたモデル構築, 125
サンプル ノード、CLEF, 35
サーバー
  構成オプション、デバッグの変更, 277
  ディレクトリ設定コントロール, 194
  一時ファイル, 76
  ファイル設定コントロール, 195
サーバー側
  コンポーネント, 2
  ライブラリ, 48, 78, 252
サーバー側 API, 5, 242
  使用, 268
  機能, 250
  アーキテクチャ, 242
予測サーバー API (PSAPI), 242
サービス関数、API, 242-243

システム
  コントロール, 170
  メニュー, 150
ショートカット
  CLEF の, 56, 157

隠す
  マイニング機能、モデル ビルダー, 110
スクリプト名, 66
  ノードの指定, 69, 103
  プロパティの指定, 72
ステータス領域、ダイアログ ボックス, 29
ストレージ タイプ, 251
スピン コントロール, 199

ダイアログ ボックス、デザイン, 23
タイトル バー、ダイアログ ボックス, 27
タブ領域、ダイアログ ボックス, 29
タブ、ダイアログまたはウィンドウでの定義, 155

チェック ボックス, 178
チェック ボックス グループ, 179
  表示順の変更, 207
  行数の変更, 206
行、チェック ボックスおよびラジオ ボタン グループの数の変更, 206

チャンネル関数、API, 247

ツールバー
  項目、カスタム, 17, 153
  領域、ダイアログ ボックス, 28
ツールヒント テキスト、指定, 31, 55

テキスト
  領域コントロール, 202
  ブラウザ パネル, 159
  ボックス コントロール, 203
静的テキスト, 169
テスト
  CLEF の拡張, 274
  ローカライズされたノードとヘルプ, 237
デバッグ
  拡張, 275
  サーバー構成オプションの変更, 277
  [デバッグ] タブ、ノードのダイアログ, 45, 276
データ モデル, 4, 258
  処理, 253
  プロバイダ, 94
データ モデル ドキュメント、XML 出力, 258
データのスコアリング, 32
データベース
  接続設定, 183
  テーブル設定, 185
テーブル コントロール, 200

ドキュメント, 54, 108
  構築, 135
  出力オブジェクト, 14
  タイプ, 54
  出力、ノードの定義, 136
  ビルダー ノード, 14, 38, 67, 108, 135
実行要件ドキュメント、XML 出力, 261
状況詳細ドキュメント、XML 出力, 264

ナゲット、モデル, 12
正確なコントロールの位置、指定, 208

法律に関する注意事項, 443

拡張の配布, 278
構造の宣言, 83
拡張のアンインストール, 279
拡張のインストール, 279
以前のバージョンとの互換性、拡張の保持, 105
値の種類、プロパティ, 83

値のリスト, 56
値のリスト、列挙型プロパティによる使用, 84
ノード
  関数、API, 246
  アイコン、デザイン, 18
  名前、カスタム, 31
  キャッシュ ステータス, 19
  タイプ, 4, 250
  情報ドキュメント (XML), 252
ノード情報ドキュメント、XML 出力, 262

パスワード ボックス, 190
パネル
  設定, 159
  拡張オブジェクト, 162
  テキスト ブラウザ, 159
  プロパティ サブパネル, 172
  プロパティ パネル, 164, 172
  モデル ビューア, 166
パネル領域、ダイアログ ボックス, 29
パラメータ ドキュメント、XML 出力, 262
パレット
  非表示, 64
  削除, 64
  ノードの指定, 17, 59, 67
パレットおよびサブパレットの非表示, 64
ハンドル、コールバック関数, 245

ピア, 242
  関数、API, 244

ファイル構造, 6
反復、特性ファイル, 92
操作、特性ファイル, 88
条件、特性ファイル, 98
  単純, 102
  複合, 103
  画面のコンポーネントの表示を制御するために使用, 222
  表示特性の制御のために使用, 217
一時ファイル, 252
  サーバー, 76
入力ファイル, 5, 76
特性ファイル, 1, 4, 42
ファイルスペース, 252
入力ファイルの ContainerFile 要素、特性ファイル, 77
出力ファイルの ContainerFile 要素、特性ファイル, 78
フィールド
  groups, 117-118
  セット, 95
  メタデータ, 94
複数フィールド設定, 186
単一フィールド設定コントロール, 196
フォルダ、拡張, 6
フレーム クラス, 147
プロセス フロー、サーバー側 API, 247
拡張プロセスの外部実行, 275
プロバイダ、データ モデル, 94
プロパティ
  列挙型, 83
  定義, 71
  キー, 51, 86
  サブパネル, 172
  の設定の調査, 276
  パネル, 164
  パネル (ネスト), 175
  ランタイム, 76
プロパティ設定、調査, 276
列挙型プロパティ, 83-84
構造化プロパティ, 85
プロパティ コントロール, 167
  PropertyControl 要素, 191
  UI コンポーネント, 168
  コントローラ, 175
  プロパティ パネル, 172
プロパティ コントロールのカスタム レイアウト, 206
  単純, 206
  詳細, 207
プロパティ コントロールのレイアウト
  標準, 204
  カスタム, 206
プロパティ パネル コントロール
  プロパティ サブパネル, 172
  プロパティ パネル (ネスト), 175
プロパティ ファイル (.properties), 231
プロパティ、タイプ
  列挙型, 84
  構造化, 85
プロパティ、データ モデルの指定, 89, 97

ヘルプ システム
  場所, 226
  リンク先, 224
  ローカライズ, 236
ヘルプ トピック、表示の指定, 227
ヘルプ リンク、ノードの指定, 69
ヘルプセット ファイル、JavaHelp, 225

ホスト関数、APIs, 246
ホスト情報ドキュメント、XML 出力, 261
ボタン領域、ダイアログ ボックス, 32

マイニング機能、モデル ビルダー, 110

調整済み傾向, 89

メイン ウィンドウ、カスタマイズ, 154
メタデータ、フィールド, 94
メニュー、標準およびカスタム, 16, 150

モジュール、拡張, 242
モジュール関数、API, 244
モジュール情報ドキュメント、XML 出力, 262
モデル, 107
  自動化, 125
  構築, 109
  適用, 134
  data, 4
  interactive, 120
  ビューア パネル, 166
モデル出力
  オブジェクト, 12, 108
  ノードの定義, 119
役割、モデル出力, 97
署名、モデル, 116
複製、モデル, 53
[モデル] タブ、マネージャ領域, 119
モデルの適用, 134

ユーザー インターフェイス
  定義, 142
  デザイン, 23

共有ライブラリ, 48, 78, 252
ライブラリ、共有（サーバー側）, 48, 78, 252
ラジオ ボタン グループ, 192
  表示順の変更, 207
  行数の変更, 206
ラベル、コンポーネントの上の配置, 206
ランタイム プロパティ, 76

リソース、拡張, 242
リソース バンドル, 47

レイアウト、プロパティ コントロール
  標準, 204
  カスタム, 206

ロケール、Windows の設定, 230

ローカライズ
  拡張, 230
  エラー メッセージ, 265
  ヘルプ システム, 236