

*IBM SPSS Modeler の拡張機能*



## 注記

本書および本書で紹介する製品をご使用になる前に、[57 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® SPSS® Modeler バージョン 18 リリース 2 モディフィケーション 2 および新しい版で明記されない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

**原典：**

IBM SPSS Modeler Extensions

**発行：**

日本アイ・ビー・エム株式会社

**担当：**

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation .

# 目次

<b>第 1 章サポートされる言語.....</b>	<b>1</b>
R.....	1
Python for Spark.....	1
Python for Spark を使用したスクリプト.....	2
拡張ノード.....	12
拡張のエクスポート・ノード.....	12
拡張の出力ノード.....	14
拡張モデル・ノード.....	16
拡張モデル・ナゲット.....	17
拡張の変換ノード.....	19
拡張のインポート・ノード.....	20
<b>第 2 章拡張機能.....</b>	<b>23</b>
拡張ハブ.....	23
「探索」タブ.....	24
「インストール済み」タブ.....	25
設定.....	25
拡張機能の詳細.....	25
ローカルの拡張バンドルのインストール.....	26
拡張のインストール場所.....	26
必須の R パッケージ.....	27
カスタム・ノードの作成と管理.....	27
カスタム・ダイアログ・ビルダーのレイアウト.....	28
カスタム・ノード・ダイアログの作成.....	28
ダイアログのプロパティ.....	29
ダイアログ・キャンバスのコントロールのレイアウト.....	29
スクリプト・テンプレートの作成.....	30
カスタム・ノード・ダイアログのプレビュー.....	31
コントロール・タイプ.....	32
拡張プロパティ.....	50
カスタム・ノード・ダイアログの管理.....	52
ローカライズされたカスタム・ノード・ダイアログの作成.....	54
Python for Spark を使用したデータのインポートおよびエクスポート.....	56
R を使用したデータのインポートおよびエクスポート.....	56
<b>特記事項.....</b>	<b>57</b>
商標.....	58
製品資料に関するご使用条件.....	58
<b>索引.....</b>	<b>61</b>



# 第1章 サポートされる言語

IBM SPSS Modeler は、R と Apache Spark をサポートします (Python 経由)。詳しくは、以下のセクションを参照してください。

## R

IBM SPSS Modeler は R をサポートします。

### 利用可能なシンタックス

- さまざまな拡張ノードの「シンタックス」タブにあるシンタックス・フィールド内では、R によって認識されるステートメントおよび関数のみが許容されます。
- 拡張の変換ノードおよび拡張モデル・ナゲットの場合、データは R スクリプトを使用して (バッチで) 渡されます。このため、モデル・スコアリングおよびプロセス・ノード用の R スクリプトには、ソートや集計など、データ内の複数の行にまたがる操作や、複数の行を結合する操作を含めることはできません。この制約により、Hadoop 環境や、データベース内マイニングの間に、確実にデータが分割された状態になるよう強制することができます。拡張の出力ノードおよび拡張モデル構築ノードには、この制約はありません。
- 拡張の変換ノードと拡張モデル・ナゲットのいずれの場合も、非バッチ・データ転送モードの追加により、SPSS Modeler Server で、データ内の行をまたいだり組み合わせたりできるようになりました。
- すべての R ノードを独立したグローバルな R 環境とみなすことができます。したがって、2 つの別個の R ノード内で `library` 関数を使用する場合、両方の R スクリプトで R ライブラリをロードする必要があります。
- R スクリプト内に定義されている R オブジェクトの値を表示するには、`printing` 関数への呼び出しを含める必要があります。例えば、`data` という名前の R オブジェクトの値を表示するには、R スクリプトに次の行を含めます。

```
print(data)
```

- R スクリプトに、R の `setwd` 関数への呼び出しを含めることはできません。この関数は、IBM SPSS Modeler が R スクリプトの出力ファイルのファイル・パス制御に使用するためです。
- CLEM の式およびスクリプトで使用するために定義されるストリーム・パラメーターは、R スクリプトで使用しても認識されません。
- IBM SPSS Modeler は R のインタラクティブ プロットをサポートしていません。

## Python for Spark

IBM SPSS Modeler は、Apache Spark 用の Python スクリプトをサポートします。

注:

- Python ノードは Spark 環境に依存します。
- Python スクリプトでは、データが Spark DataFrame 形式で表示されるため、Spark API を使用する必要があります。
- バージョン 17.1 で作成された古いノードは、引き続き IBM SPSS Analytic Server に対してのみ実行されます (データは IBM SPSS Analytic Server 入力ノードから発信され、IBM SPSS Modeler サーバーには抽出されていません)。バージョン 18.0 以降で作成された新しい Python およびカスタム・ダイアログ・ビルダーのノードは IBM SPSS Modeler サーバーに対して実行できます。
- Python をインストールする際に、必ずすべてのユーザーに Python インストール済み環境にアクセスする許可を付与してください。

- Machine Learning Library (MLlib) を使用する場合、NumPy を含む Python のバージョンをインストールする必要があります。その後、Python インストール済み環境を使用するように IBM SPSS Modeler サーバー (または IBM SPSS Modeler クライアントのローカル・サーバー) を構成する必要があります。詳しくは、2 ページの『[Python for Spark を使用したスクリプト](#)』を参照してください。

## Python for Spark を使用したスクリプト

IBM SPSS Modeler は、データを処理するために Apache Spark フレームワークを使用しながら Python スクリプトを実行することができます。ここでは、付属するインターフェース用の Python API について説明します。

IBM SPSS Modeler インストールには、Spark ディストリビューションが含まれています (例えば、IBM SPSS Modeler 18.2.2 には、Spark 2.4.6 が含まれています)。

### 前提条件

- Python/Spark スクリプトを IBM SPSS Analytic Server に対して実行するには、Analytic Server に接続する必要があります。また、Analytic Server が、互換性のある Apache Spark のインストール済み環境にアクセスできる状態になっている必要があります。Apache Spark を実行エンジンとして使用する方法については、IBM SPSS Analytic Server の資料を参照してください。
- Python/Spark スクリプトを IBM SPSS Modeler Server (または Windows 64 または Mac64 を必要とする IBM SPSS Modeler クライアントに含まれるローカルサーバー) に対して実行する場合、Python をインストールしたり、その Python インストール済み環境を使用するために options.cfg を編集したりする必要はなくなりました。バージョン 18.1 以降では、IBM SPSS Modeler に Python ディストリビューションが含まれるようになりました。ただし、デフォルトの IBM SPSS Modeler Python ディストリビューションに含まれない特定のモジュールが必要な場合は、<Modeler\_installation\_directory>/python に移動し、追加のパッケージをインストールできます。

IBM SPSS Modeler に Python ディストリビューションが含まれるようになりましたが、以前のリリースと同様に、必要に応じて options.cfg に次のオプションを追加することにより、独自の Python インストール済み環境を指定できます。

```
# Set to the full path to the python executable
(including the executable name) to enable use of PySpark.
eas_pyspark_python_path, ""
```

Windows の例:

```
eas_pyspark_python_path, "C:¥¥Your_Python_Install¥¥python.exe"
```

Linux の例:

```
eas_pyspark_python_path, "/Your_Python_Install/bin/python"
```

注: 独自の Python インストール済み環境を指定する場合、バージョン 3.7.x である必要があります。IBM SPSS Modeler は Anaconda 3 および Python 3.7.6 でテスト済みです。

### IBM SPSS Analytic Server のコンテキスト オブジェクト

Python/Spark スクリプトの実行コンテキストは、Analytic Server のコンテキスト オブジェクトによって定義されます。IBM SPSS Modeler サーバーに対して実行する場合、このコンテキスト・オブジェクトは、IBM SPSS Modeler サーバーのインストールに含まれる組み込みバージョンの Analytic Server に対応します。このコンテキスト オブジェクトを取得するには、スクリプトに以下のコードが記述されている必要があります。

```
import spss.pyspark.runtime
asContext = spss.pyspark.runtime.getContext()
```

以下のように指定すると、Analytic Server コンテキストから Spark コンテキストと SQL コンテキストを取得することができます。

```
sparkContext = asc.getSparkContext()
sqlContext = asc.getSparkSQLContext()
```

Spark コンテキストと SQL コンテキストについては、Apache Spark の資料を参照してください。

## データへのアクセス

データは、Spark SQL DataFrame の形式で、Python/Spark スクリプトと実行コンテキスト間で転送されます。データを消費するスクリプト (つまり、入力ノードを除くすべてのノード) では、以下のように、コンテキストからデータ フレームを取得する必要があります。

```
inputData = asContext.getSparkInputData()
```

データを生成するスクリプト (つまり、ターミナル ノードを除くすべてのノード) では、以下のように、データ フレームをコンテキストに返す必要があります。

```
asContext.setSparkOutputData(outputData)
```

必要な場合は、以下のように SQL コンテキストを使用して、RDD から出力データ フレームを作成することができます。

```
outputData = sqlContext.createDataFrame(rdd)
```

## データ モデルの定義

データを生成するノードでは、そのノードの下流で表示されるフィールドを記述するデータ モデルも定義する必要があります。Spark SQL の用語では、データ モデルはスキーマと呼ばれます。

Python/Spark スクリプトでは、`pyspark.sql.types.StructType` オブジェクトの形式で出力データ モデルが定義されます。`StructType` は、`StructField` オブジェクトのリストから構成され、出力データ フレーム内の行を記述します。それぞれの `StructField` により、出力データ モデル内の 1 つのフィールドが記述されます。

以下のように、入力データ フレームの `:schema` 属性を使用して、入力データのデータ モデルを取得することができます。

```
inputSchema = inputData.schema
```

変更されることなく受け渡しされるフィールドは、入力データ モデルから出力データ モデルにコピーすることができます。以下のように `StructField` コンストラクタを使用して、出力データ モデル内の新しいフィールドや変更されるフィールドを作成することができます。

```
field = StructField(name, dataType, nullable=True, metadata=None)
```

このコンストラクタについては、Spark の資料を参照してください。

少なくとも、フィールド名とそのフィールドのデータ型を指定する必要があります。必要に応じて、フィールドの尺度、役割、説明を設定するためのメタデータを指定することができます ([7 ページの『データのメタデータ』](#)を参照)。

## DataModelOnly モード

下流での編集を有効にするには、特定のノードを実行する前に、そのノードの出力データ モデルを IBM SPSS Modeler に認識させる必要があります。使用可能なデータがない場合、IBM SPSS Modeler は、Python/Spark ノードの出力データ モデルを取得するために、データ モデル専用モードという特殊なモードでスクリプトを実行します。Analytic Server コンテキスト オブジェクトで `isComputeDataModelOnly` メソッドを使用すると、スクリプトでこの特殊なモードを識別することができます。

変換ノードのスクリプトでは、以下のような一般的なパターンを使用することができます。

```
if asContext.isComputeDataModelOnly():
    inputSchema = asContext.getSparkInputSchema()
```

```
outputSchema = ... # construct the output data model
asContext.setSparkOutputSchema(outputSchema)
else:
    inputData = asContext.getSparkInputData()
    outputData = ... # construct the output data frame
    asContext.setSparkOutputData(outputData)
```

## モデルの作成

モデルを作成するノードは、そのモデルを適切に記述する何らかのコンテンツを実行コンテキストに返す必要があります。これにより、そのモデルを適用するノードが、後でそのモデルを正確に再作成できるようになります。

モデルのコンテンツは、キーと値のペアという観点から定義されます。この場合、キーと値の意味はビルドノードとスコアノードでのみ認識され、Modeler がキーと値の意味を解釈することはありません。必要な場合は、ノードで MIME タイプを特定の値に割り当て、ユーザーが認識できるタイプの値を Modeler のモデルナゲットに表示することができます。

このコンテキストの値は、PMML、HTML、画像などになります。モデルコンテンツに値を追加するには、ビルドスクリプトで以下のように指定します。

```
asContext.setModelContentFromString(key, value, mimeType=None)
```

モデルコンテンツから値を取得するには、スコアスクリプトで以下のように指定します。

```
value = asContext.getModelContentToString(key)
```

モデル全体またはモデルの一部がファイルシステム内のファイルまたはフォルダに保管されている場合は、簡単な方法として、そのファイルまたはフォルダに保管されているすべてのコンテンツを 1 回の呼び出しでバンドルすることができます。これを行うには、ビルドスクリプトで以下のように指定します。

```
asContext.setModelContentFromPath(key, path)
```

この場合、MIME タイプを指定するオプションはないことに注意してください。これは、バンドルにさまざまなコンテンツタイプが含まれている可能性があるためです。

モデルの作成時にコンテンツを保管するための一時的な場所が必要な場合は、以下のように指定すると、適切な場所をコンテキストから取得することができます。

```
path = asContext.createTemporaryFolder()
```

既存のコンテンツをファイルシステム内の一時的な場所に取り込むには、スコアスクリプトで以下のように指定します。

```
path = asContext.getModelContentToPath(key)
```

## エラー処理

エラーを発生させるには、スクリプトから例外をスローし、その例外を IBM SPSS Modeler ユーザーに対して表示します。一部の例外については、`spss.pyspark.exceptions` モジュールで事前に定義されています。以下に例を示します。

```
from spss.pyspark.exceptions import ASContextException
if ... some error condition ...:
    raise ASContextException("message to display to user")
```

## Analytic Server コンテキスト

Analytic Server コンテキストにより、IBM SPSS Analytic Server との対話を行うための Analytic Server のコンテキストインターフェースがサポートされます。



## AnalyticServerContext オブジェクト

AnalyticServerContext オブジェクトは、IBM SPSS Analytic Server との対話を行うための複数のインターフェースを提供するコンテキスト環境を設定します。このコンテキスト インスタンスを作成する必要があるアプリケーションでは、`spss.pyspark.runtime.getContext()` インターフェースを直接実装するのではなく、`spss.pyspark.runtime.getContext()` インターフェースを使用してコンテキスト インスタンスを作成する必要があります。

以下のように指定すると、Pyspark Python SparkContext インスタンスが返されます。

```
cxt.getSparkContext() : SparkContext
```

以下のように指定すると、Pyspark Python SQLContext インスタンスが返されます。

```
cxt.getSparkSQLContext() : SQLContext
```

以下のように指定すると、出力データ モデルを計算するためだけに操作が実行された場合は `True` が返されます。それ以外の場合は `False` が返されます。

```
cxt.isComputeDataModelOnly() : Boolean
```

以下のように指定すると、Spark 環境内でスクリプトが実行されている場合は `True` が返されます。現時点では、常に `True` が返されます。

```
cxt.isSparkExecution() : Boolean
```

以下のように指定すると、上流の一時ファイルから入力データが読み込まれ、`pyspark.sql.DataFrame` インスタンスが生成されます。

```
cxt.getSparkInputData() : DataFrame
```

以下のように指定すると、入力データ モデルから生成された `pyspark.sql.StructType` インスタンスが返されます。入力データ モデルが存在しない場合は `None` が返されます。

```
cxt.getSparkInputSchema() : StructType
```

以下のように指定すると、Analytic Server コンテキストに出力データ フレームが直列化され、コンテキストが返されます。

```
cxt.setSparkOutputData( outDF) : AnalyticServerContext
```

パラメーター:

- `outDF (DataFrame)` : 出力データ フレームの値

例外:

- `DataOutputNotSupported` : このインターフェースが `pyspark:buildmodel` 関数で呼び出された場合にスローされます。
- `ASContextException` : 出力データ フレームが `None` の場合にスローされます。
- `InconsistentOutputDataModel` : 対象となる両方のオブジェクトに共通するフィールド名とストレージタイプの情報が矛盾している場合にスローされます。

以下のように指定すると、`outSchema StructType` インスタンスがデータ モデルに変換され、そのデータ モデルが Analytic Server コンテキストに直列化され、コンテキストが返されます。

```
cxt.setSparkOutputSchema(outSchema) : AnalyticServerContext
```

パラメーター:

- `outSchema(StructType)` : 出力 `StructType` オブジェクト

例外:

- `ASContextException` : 出力スキーマ インスタンスが `None` の場合にスローされます。
- `InconsistentOutputDataModel` : 対象となる両方のオブジェクトに共通するフィールド名とストレージタイプの情報が矛盾している場合にスローされます。

以下のように指定すると、モデルを作成するための出力情報の場所が `Analytic Server` コンテキストに保管され、コンテキストが返されます。

```
cxt.setModelContentFromPath(key, path, mimetype=None) : AnalyticServerContext
```

この場合のパスは、生成される `cxt.createTemporaryFolder()` API を使用するディレクトリのパスにすることができます (このディレクトリ内のすべてのものがモデル コンテンツとしてパッケージ化されている場合)。

パラメーター:

- `key` (`string`) : キーの文字列値
- `path` (`string`) : モデル作成用の出力の場所の文字列パス
- `mimetype` (`string, optional`) : コンテンツの MIME タイプ

例外:

- `ModelOutputNotSupported` : この API が `pyspark:buildmodel` 関数から呼び出されなかった場合にスローされます。
- `KeyError` : キー属性が `None` の場合や、文字列が空の場合にスローされます。

以下のように指定すると、モデル作成用コンテンツ、メタデータ、その他の属性が `Analytic Server` コンテキストに格納され、コンテキストが返されます。

```
cxt.setModelContentFromString(key, value, mimetype=None) : AnalyticServerContext
```

パラメーター:

- `key` (`string`) : キーの文字列値
- `value` (`string`) : モデル メタデータの文字列値
- `mimetype` (`string, optional`) : コンテンツの MIME タイプ

例外:

- `ModelOutputNotSupported` : この API が `pyspark:buildmodel` 関数から呼び出されなかった場合にスローされます。
- `KeyError` : キー属性が `None` の場合や、文字列が空の場合にスローされます。

以下のように指定すると、`Analytic Server` が管理する一時フォルダの場所が返されます。この一時フォルダを使用して、モデルのコンテンツを保管することができます。

```
cxt.createTemporaryFolder() : string
```

例外:

- `ModelOutputNotSupported` : この API が `pyspark:buildmodel` 関数から呼び出されなかった場合にスローされます。

以下のように指定すると、入力キーに一致するモデルの場所が返されます。

```
cxt.getModelContentToPath(key) : string
```

パラメーター:

- `key` (`string`) : キーの文字列値

例外:

- `ModelInputNotSupported` : この API が `pyspark:applymodel` 関数から呼び出されなかった場合にスローされます。
- `KeyError` : キー属性が `None` の場合や、文字列が空の場合にスローされます。
- `IncompatibleModelContentType` : モデルのコンテンツ タイプがコンテナではない場合にスローされます。

以下のように指定すると、入力キーに一致するモデルのコンテンツ、モデルのメタデータ、その他のモデル属性が返されます。

```
cxt.getModelContentToString(key) : string
```

パラメーター:

- `key (string)` : キーの文字列値

例外:

- `ModelInputNotSupported` : この API が `pyspark:applymodel` 関数から呼び出されなかった場合にスローされます。
- `KeyError` : キー属性が `None` の場合、文字列が空の場合、またはキーが存在しない場合にスローされます。
- `IncompatibleModelContentType` : モデルのコンテンツ タイプが矛盾している場合にスローされます。

以下のように指定すると、入力キーに割り当てられている MIME タイプが返されます。指定されたコンテンツに MIME タイプが含まれていない場合は、`None` が返されます。

```
cxt.getModelContentMimeType(key) : string
```

パラメーター:

- `key (string)` : キーの文字列値

例外:

- `ModelInputNotSupported` : この API が `pyspark:applymodel` 関数から呼び出されなかった場合にスローされます。
- `KeyError` : キー属性が `None` の場合、文字列が空の場合、またはキーが存在しない場合にスローされます。

## データのメタデータ

このセクションでは、`pyspark.sql.StructField` に基づいてデータ モデルの属性を設定する方法について説明します。

### `spss.datamodel.Role` オブジェクト

このクラスは、データ モデル内の各フィールドの有効な役割を列挙します。

**BOTH:** このフィールドは、前提条件にすることも、結果にすることもできます。

**FREQWEIGHT:** このフィールドは、度数の重みとして使用されます。これは、ユーザーには表示されません。

**INPUT:** このフィールドは、予測値または前提条件のいずれかになります。

**NONE:** モデルの作成時に、このフィールドが直接使用されることはありません。

**TARGET:** このフィールドは、予測値または結果のいずれかになります。

**PARTITION:** このフィールドを使用して、データ区分が識別されます。

**RECORDID:** このフィールドを使用して、レコード ID が識別されます。

SPLIT: このフィールドを使用して、データが分割されます。

### **spss.datamodel.Measure オブジェクト**

このクラスは、データ モデル内の各フィールドの尺度を列挙します。

UNKNOWN: 不明な型の尺度であることを示します。

CONTINUOUS: 連続型の尺度であることを示します。

NOMINAL: 名義型の尺度であることを示します。

FLAG: フィールド値が2つの値のいずれかであることを示します。

DISCRETE: フィールド値を値のセットとして解釈する必要があることを示します。

ORDINAL: 順序型の測定タイプであることを示します。

TYPELESS: フィールドのストレージに対応する値であれば、任意の値をそのフィールドに設定することができます。

### **pyspark.sql.StructField オブジェクト**

StructType 内のフィールドを表します。StructField オブジェクトは、以下に示す4つのフィールドから構成されます。

- name (string): StructField の名前
- dataType (pyspark.sql.DataType): 特定のデータ型
- nullable (bool): StructField の値に None の値を格納できるかどうか
- metadata (dictionary): オプションの属性を保管するための Python 辞書

メタデータの辞書インスタンスを使用すると、特定のフィールドについて、尺度、役割、ラベルの属性を保管することができます。これらの属性のキーワードを以下に示します。

- measure: measure 属性のキーワード
- role: role 属性のキーワード
- displayLabel: label 属性のキーワード

例:

```
from spss.datamodel.Role import Role
from spss.datamodel.Measure import Measure
_metadata = {}
_metadata['measure'] = Measure.TYPELESS
_metadata['role'] = Role.NONE
_metadata['displayLabel'] = "field label description"
StructField("userName", StringType(), nullable=False,
metadata=_metadata)
```

### **日付、時刻、タイムスタンプ**

日付型、時刻型、またはタイムスタンプ型のデータを使用する操作の場合、1970-01-01:00:00:00 という値 (協定世界時を使用) に基づいて、データの値が実際の値に変換されます。

日付データの場合、データの値は、1970-01-01 という値 (協定世界時を使用) に基づく日数を表します。

時刻データの場合、データの値は 24 時間の範囲内における秒数を表します。

タイムスタンプデータの場合、データの値は、1970-01-01:00:00:00 という値 (協定世界時を使用) に基づく秒数を表します。

### **例外**

このセクションでは、発生する可能性がある例外インスタンスについて説明します。

### **MetadataException オブジェクト**

Python 例外のサブクラス。

この例外は、メタデータ オブジェクトの操作でエラーが発生した場合にスローされます。

### **UnsupportedOperationException オブジェクト**

Python 例外のサブクラス。

この例外は、許可されていない特定の操作が実行された場合にスローされます。

### **InconsistentOutputDataModel オブジェクト**

Python 例外のサブクラス。

この例外は、`setSparkOutputSchema` オブジェクトと `setSparkOutputData` オブジェクトの両方が呼び出され、両方のオブジェクトに共通するフィールド名とストレージタイプの情報が矛盾している場合にスローされます。

### **IncompatibleModelContentType オブジェクト**

Python 例外のサブクラス。

この例外は、以下の場合にスローされます。

- `setModelContentFormString` を使用してモデルを設定し、`getModelContentToPath` を使用して値を取得した場合
- `setModelContentFormPath` を使用してモデルを設定し、`getModelContentToString` を使用して値を取得した場合

### **DataOutputNotSupported オブジェクト**

Python 例外のサブクラス。

この例外は、`pyspark:buildmodel` 関数で操作を実行した場合に `setSparkOutputData` オブジェクト内で発生します。

### **ModelInputNotSupported オブジェクト**

Python 例外のサブクラス。

この例外は、スクリプトが `pyspark:applymodel` 関数内で `getModelContentPathByKey` API と `getModelContentToString` API を呼び出さなかった場合にのみ発生します。

### **ModelOutputNotSupported オブジェクト**

Python 例外のサブクラス。

この例外は、スクリプトが `pyspark:buildmodel` 関数内で `setModelContentFromPath` API と `setModelContentFromString` API を呼び出さなかった場合にのみ発生します。

### **ASContextException オブジェクト**

Python 例外のサブクラス。

この例外は、予期しない実行時例外が発生した場合にスローされます。

### **例**

このセクションでは、Python for Spark のスクリプトの例を紹介します。

## データを処理するための基本的なスクリプトの例

```
import spss.pyspark.runtime
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = cxt.getSparkInputSchema()
    cxt.setSparkOutputSchema(_schema)
else:
    _structType = cxt.getSparkInputSchema()
    df = cxt.getSparkInputData()
    _newDF = df.sample(False, 0.01, 1)
    cxt.setSparkOutputData(_newDF)
```

## LinearRegressionWithSGD アルゴリズムを使用するモデル作成スクリプトの例

```
from pyspark.context import SparkContext
from pyspark.sql.context import SQLContext
from pyspark.sql import Row
from pyspark.mllib.regression import
    LabeledPoint, LinearRegressionWithSGD, LinearRegressionModel
from pyspark.mllib.linalg import DenseVector
import numpy
import json

import spss.pyspark.runtime
from spss.pyspark.exceptions import ASContextException

ascontext = spss.pyspark.runtime.getContext()
sc = ascontext.getSparkContext()
df = ascontext.getSparkInputData()

# field settings amd algorithm parameters

target = '%%target_field%%'
predictors = [%%predictor_fields%%]
num_iterations=%%num_iterations%%
prediction_field = "$LR-" + target

# save linear regression model to a filesystem path

def save(model, sc, path):
    data =
sc.parallelize([json.dumps({"intercept":model.intercept,"weights":model.weights.tolist()})])
    data.saveAsTextFile(path)

# print model details to stdout

def dump(model,predictors):
    print(prediction_field+" = " + str(model.intercept))
    weights = model.weights.tolist()
    for i in range(0,len(predictors)):
        print("%t+ "+predictors[i]+"*"+ str(weights[i]))

# check that required fields exist in the input data

input_field_names = [ty[0] for ty in df.dtypes[:]]
if target not in input_field_names:
    raise ASContextException("target field "+target+" not found")
for predictor in
predictors:
    if predictor not in input_field_names:
        raise ASContextException("predictor field "+predictor+" not found")

# define map function to convert from dataframe Row objects to mllib LabeledPoint

def row2LabeledPoint(target,predictors,row):
    pvals = []
    for predictor in predictors:
        pval = getattr(row,predictor)
        pvals.append(float(pval))
    tval = getattr(row,target)
    return LabeledPoint(float(tval),DenseVector(pvals))

# convert dataframe to an RDD containing LabeledPoint

training_points = df.rdd.map(lambda row:
row2LabeledPoint(target,predictors,row))
```

```

# build the model
model = LinearRegressionWithSGD.train(training_points,num_iterations,intercept=True)

# write a text description of the model to stdout
dump(model,predictors)

# save the model to the filesystem and store into the output model content

modelpath = ascontext.createTemporaryFolder()
save(model,sc,modelpath)
ascontext.setModelContentFromPath("model",modelpath)

```

## LinearRegressionWithSGD アルゴリズムを使用するモデル スコアリング スクリプトの例

```

import json
import spss.pyspark.runtime
from pyspark.sql import Row
from pyspark.mllib.regression import
LabeledPoint,LinearRegressionWithSGD, LinearRegressionModel
from pyspark.mllib.linalg import DenseVector
from pyspark.sql.context import SQLContext
import numpy
from pyspark.sql.types import DoubleType, StructField

ascontext = spss.pyspark.runtime.getContext()
sc = ascontext.getSparkContext()

prediction_field = "$LR-" + '%%target_field%%'
predictors = [%%predictor_fields%%]

# compute the output schema by adding the prediction field
outputSchema = ascontext.getSparkInputSchema()
outputSchema.fields.append(StructField(prediction_field,
DoubleType(), nullable=True))

# make a prediction based on a regression model and Dataframe Row object
# return a list containing the input row values and the predicted value
def predict(row,model,predictors,infields,prediction_field_name):
    pvals = []
    rdict = row.asDict()
    for predictor in predictors:
        pvals.append(float(rdict[predictor]))
    estimate = float(model.predict(pvals))
    result = []
    for field in infields:
        result.append(rdict[field])
    result.append(estimate)
    return result

# load a serialized model from the filesystem

def load(sc, path):
    js = sc.textFile(path).take(1)[0]
    obj = json.loads(js)
    weights = numpy.array(obj["weights"])
    intercept = obj["intercept"]
    return LinearRegressionModel(weights,intercept)

ascontext.setSparkOutputSchema(outputSchema)

if not ascontext.isComputeDataModelOnly():
    # score the data in the input data frame
    indf = ascontext.getSparkInputData()

    model_path = ascontext.getModelContentToPath("model")
    model = load(sc,model_path)

    # compute the scores
    infield_names = [ty[0] for ty in indf.dtypes[:]]
    scores_rdd = indf.rdd.map(lambda
row:predict(row,model,predictors,infield_names,prediction_field))

    # create an output DataFrame containing the scores
    sqlCtx = SQLContext(sc)
    outdf = sqlCtx.createDataFrame(scores_rdd,schema=outputSchema)

```

```
# return the output DataFrame as the result
ascontext.setSparkOutputData(outdf)
```

## 拡張ノード

IBM SPSS Modeler およびそのデータ・マイニング機能を補足するために、拡張ノードでは、エキスパート・ユーザーが独自の R スクリプトまたは Python for Spark スクリプトを入力して、データ処理、モデル作成およびモデル・スコアリングを実行できるようにしています。

### 拡張のエキスポート・ノード

拡張のエキスポート・ノードを使用すると、R スクリプトまたは Python for Spark スクリプトを実行して、データをエキスポートできます。

#### 拡張のエキスポート・ノード - 「シンタックス」 タブ

シンタックスのタイプ (**R** または **Python for Spark**) を選択します。詳しくは、以下のセクションを参照してください。シンタックスの準備ができたなら、「実行」をクリックして、拡張のエキスポート・ノードを実行できます。

#### R シンタックス

「**R シンタックス**」。データ分析用のカスタムの R スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。

「**フラグ型フィールドの変換**」。フラグ型フィールドの処理方法を指定します。「**文字列を因子に、整数および実数を倍精度に**」および「**論理値(真、偽)**」の 2 つのオプションがあります。「**論理値(真、偽)**」を選択した場合、フラグ型フィールドの元の値は失われます。例えば、フィールドに「Male」および「Female」という値がある場合、これらの値は「真」および「偽」に変更されます。

「**欠損値を R の欠損値 (NA) に変換**」。選択すると、欠損値はすべて、R の NA 値に変換されます。R では、欠損値の識別に値 NA が使用されます。使用する R 関数によっては、データに NA が含まれていた場合の関数の動作を制御するために使用される引数が含まれている場合があります。例えば、関数によって NA を含むレコードを自動的に除外することを選択できる場合があります。このオプションが選択されない場合、すべての欠損値はそのまま R に渡されます。これらの欠損値は R スクリプトの実行時にエラーの原因となる可能性があります。

「**時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換**」。選択すると、日付形式または日付/時刻形式の変数が R の日付/時刻形式に変換されます。次のいずれかのオプションを選択する必要があります。

- 「**R POSIXct**」。日付形式または日付/時刻形式の変数が R の POSIXct オブジェクトに変換されます。
- 「**R POSIXlt (リスト)**」。日付形式または日付/時刻形式の変数が R の POSIXlt オブジェクトに変換されます。

注：POSIX 形式は、拡張オプションです。これらのオプションは、ご使用の R スクリプトで、これらの形式を必要とする方法で日付/時刻フィールドを処理するように指定している場合にのみ使用してください。POSIX 形式は、時刻形式の変数には適用されません。

#### Python シンタックス (Python Syntax)

「**Python シンタックス (Python Syntax)**」。データ分析用のカスタムの Python スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。Python for Spark について詳しくは、[Python for Spark](#) および [Python for Spark を使用したスクリプト](#) を参照してください。

#### 拡張のエキスポート・ノード - 「コンソール出力」 タブ

「**コンソール出力**」タブには、「シンタックス」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、「シンタックス」タブの「**R シンタックス**」フィールドにある R スクリプトが実行されたときに R コンソールから受



信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトの実行時に生成される R または Python のエラー・メッセージや警告が含まれる場合があります。出力は、主にスクリプトをデバッグするために使用できます。「**コンソール出力**」タブには、「**R シンタックス**」フィールドまたは「**Python シンタックス (Python Syntax)**」フィールドのスクリプトも表示されます。

拡張エクスポート・スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「**コンソール出力**」タブが上書きされます。出力を編集することはできません。

## ストリームの公開

ストリームの公開は、標準エクスポート・ノード (データベース、フラット・ファイル、Statistics エクスポート、拡張のエクスポート、Data Collection エクスポート、SAS エクスポート、Excel、および XML エクスポートの各ノード) のいずれかを使用して、IBM SPSS Modeler から直接行われます。エクスポート・ノードのタイプにより、公開されたストリームが IBM SPSS Modeler Solution Publisher Runtime または外部アプリケーションを使用して実行されるたびに書き込まれる結果の形式が決定されます。例えば、公開されたストリームが実行されるたびに結果をデータベースに書き込む場合は、データベース・エクスポート・ノードを使用します。

## ストリームを公開するには

1. ストリームを通常の方法で開くか構築し、エクスポート・ノードを端に接続します。
2. エクスポート・ノードの「公開」タブで、公開するファイルのルート名 (.pim、.par、および .xml の拡張子が追加されるファイル名) を指定します。
3. 「公開」をクリックして、ストリームを公開するか、「**ストリームの公開**」を選択して、ノードが実行されるたびにストリームを自動的に公開するようにします。

**名前:** 公開された画像およびパラメーター・ファイルのルート名を指定します。

- **画像ファイル (\*.pim)** には、Runtime が、公開されたストリームをエクスポートの時点とまったく同じように実行するために必要なすべての情報があります。ストリームの設定 (入力データ・ソースや出力データ・ファイルなど) を変更する必要がないことが明らかな場合は、画像ファイルだけを展開できます。
- **パラメーター ファイル (\*.par)** には、データ・ソース、出力ファイル、および実行オプションに関する設定可能な情報が含まれます。ストリームを再公開せずにストリームの入力または出力を制御するには、パラメーター・ファイルおよび画像ファイルが必要です。
- **メタデータ ファイル (\*.xml)** は、画像やそのデータ・モデルの入力および出力を記述します。ランタイム ライブラリーを組み込み、入力データおよび出力データの構造を認識する必要があるアプリケーションによって使用するために設計されています。

**注:** このファイルは、「**メタデータを公開する**」オプションを選択している場合にのみ、作成されます。

**パラメーターを公開する:** 必要に応じて、\*.par ファイルにストリーム・パラメーターを含めることができます。画像を実行する場合、\*.par ファイルを編集して、またはランタイム API によって、これらのストリーム・パラメーター値を変更することができます。

このオプションを選択すると、「**パラメーター**」ボタンを使用できるようになります。このボタンをクリックすると、「**パラメーター公開**」ダイアログ・ボックスが表示されます。

「**公開**」列の関連するオプションを選択して、公開された画像に含めるパラメーターを選択します。

**ストリームの実行時:** ノードが実行されたときにストリームが自動的に公開されるかどうかを指定します。

- **データのエクスポート:** ストリームを公開せずに、標準の方法でエクスポート・ノードを実行します。(基本的に、ノードは IBM SPSS Modeler で IBM SPSS Modeler Solution Publisher が使用できない場合と同じ方法で実行します。)このオプションを選択した場合は、エクスポート・ノード・ダイアログ・ボックスで「**公開**」を明示的にクリックしない限り、ストリームは公開されません。また、ツールバーの「**公開**」ツールを使用するか、スクリプトを使用することで、現在のストリームを公開できます。
- **ストリームの公開:** IBM SPSS Modeler Solution Publisher を使用して、展開用にストリームを公開します。実行するたびにストリームを自動的に公開する場合は、このオプションを選択します。

**注:**

- 公開済みのストリームを新規データまたは更新されたデータと組み合わせて実行することを計画している場合、入力ファイル内のフィールドの順序は、公開済みのストリームで指定された入力ノードの入力ファイル内のフィールドと同じであることが必要です。
- 外部アプリケーションに公開する場合は、無関係なフィールドをフィルタリングしたり、入力条件に準拠したフィールド名に変更することを検討してください。両方とも、エクスポート・ノードの前にフィルター・ノードを使用することで達成できます。

## 拡張の出力ノード

「拡張の出力ノード」ダイアログの「出力」タブで「画面に出力」が選択されている場合は、出力ブラウザ・ウィンドウの画面上に出力が表示されます。出力は、出力マネージャーにも追加されます。出力ブラウザ・ウィンドウは、それ自体に出力を印刷または保存、または別の形式で出力をエクスポートできるメニュー・セットを備えています。「編集」メニューに含まれているのは「コピー」オプションのみです。拡張の出力ノードの出力ブラウザには、テキスト出力が表示される「テキスト出力」タブと、グラフおよび図表が表示される「グラフ出力」タブの2つのタブが表示されます。

「拡張の出力ノード」ダイアログの「出力」タブで「ファイルに出力」が選択されている場合、拡張の出力ノードの実行の成功時に出力ブラウザ・ウィンドウは表示されません。

### 拡張の出力ノードの「シンタックス」タブ

シンタックスのタイプ (**R** または **Python for Spark**) を選択します。詳しくは、以下のセクションを参照してください。シンタックスの準備ができたなら、「実行」をクリックして、拡張の出力ノードを実行できます。出力オブジェクトは出力マネージャーに追加されるか、オプションで、「出力」タブの「ファイル名」フィールドで指定されたファイルに追加されます。

#### R シンタックス

「**R シンタックス**」。データ分析用のカスタムの R スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。

「**フラグ型フィールドの変換**」。フラグ型フィールドの処理方法を指定します。「**文字列を因子に、整数および実数を倍精度に**」および「**論理値 (真、偽)**」の2つのオプションがあります。「**論理値 (真、偽)**」を選択した場合、フラグ型フィールドの元の値は失われます。例えば、フィールドに「Male」および「Female」という値がある場合、これらの値は「真」および「偽」に変更されます。

「**欠損値を R の欠損値 (NA) に変換**」。選択すると、欠損値はすべて、R の NA 値に変換されます。R では、欠損値の識別に値 NA が使用されます。使用する R 関数によっては、データに NA が含まれていた場合の関数の動作を制御するために使用される引数が含まれている場合があります。例えば、関数によって NA を含むレコードを自動的に除外することを選択できる場合があります。このオプションが選択されない場合、すべての欠損値はそのまま R に渡されます。これらの欠損値は R スクリプトの実行時にエラーの原因となる可能性があります。

「**時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換**」。選択すると、日付形式または日付/時刻形式の変数が R の日付/時刻形式に変換されます。次のいずれかのオプションを選択する必要があります。

- 「**R POSIXct**」。日付形式または日付/時刻形式の変数が R の POSIXct オブジェクトに変換されます。
- 「**R POSIXlt (リスト)**」。日付形式または日付/時刻形式の変数が R の POSIXlt オブジェクトに変換されます。

注：POSIX 形式は、拡張オプションです。これらのオプションは、ご使用の R スクリプトで、これらの形式を必要とする方法で日付/時刻フィールドを処理するように指定している場合にのみ使用してください。POSIX 形式は、時刻形式の変数には適用されません。

#### Python シンタックス (Python Syntax)

「**Python シンタックス (Python Syntax)**」。データ分析用のカスタムの Python スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。Python for Spark について詳しくは、[Python for Spark](#) および [Python for Spark を使用したスクリプト](#) を参照してください。

## 拡張の出力ノードの「コンソール出力」タブ

「コンソール出力」タブには、「シンタックス」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、「シンタックス」タブの「R シンタックス」フィールドにある R スクリプトが実行されたときに R コンソールから受信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトの実行時に生成される R または Python のエラー・メッセージや警告が含まれる場合があります。出力は、主にスクリプトをデバッグするために使用できます。「コンソール出力」タブには、「R シンタックス」フィールドまたは「Python シンタックス (Python Syntax)」フィールドのスクリプトも表示されます。

拡張の出力スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「コンソール出力」タブが上書きされます。出力を編集することはできません。

## 拡張の出力ノードの「出力」タブ

**出力名。** ノードの実行時に作成される出力の名前を指定します。「自動」を選択すると、スクリプト・タイプに応じて、出力の名前は自動的に「R 出力」または「Python 出力 (Python Output)」に設定されます。「ユーザー設定」で別の名前を指定することもできます。

**画面に出力。** 出力の生成と表示を新規ウィンドウで行うには、このオプションを選択します。出力は、出力マネージャーにも追加されます。

**ファイルに出力。** 出力をファイルに保存するには、このオプションを選択します。このオプションを選択すると、「グラフの出力」および「ファイルの出力」ラジオ・ボタンが有効になります。

**グラフの出力。** 「ファイルに出力」が選択された場合にのみ有効になります。拡張の出力ノードを実行した結果のグラフをすべてファイルに保存するには、このオプションを選択します。「ファイル名」フィールドで、生成された出力に使用するファイル名を指定してください。特定のファイルと場所を選択するには、省略記号ボタン («...») をクリックしてください。「ファイルの種類」ドロップダウン・リストでファイルの種類を指定します。次のファイルの種類を利用できます。

- 出力オブジェクト (.cou)
- HTML (.html)

**テキストの出力:** 「ファイルに出力」が選択された場合にのみ有効になります。拡張の出力ノードを実行した結果のテキスト出力をすべてファイルに保存するには、このオプションを選択します。「ファイル名」フィールドで、生成された出力に使用するファイル名を指定してください。特定のファイルと場所を指定するには、省略記号ボタン («...») をクリックしてください。「ファイルの種類」ドロップダウン・リストでファイルの種類を指定します。次のファイルの種類を利用できます。

- HTML (.html)
- 出力オブジェクト (.cou)
- テキスト文書 (.txt)

## 拡張の出力ブラウザー

「拡張の出力ノード」ダイアログ・ボックスの「出力」タブで「画面に出力」が選択されている場合は、出力ブラウザー・ウィンドウの画面上に出力が表示されます。出力は、出力マネージャーにも追加されます。出力ブラウザー・ウィンドウは、それ自体に出力を印刷または保存、または別の形式で出力をエクスポートできるメニュー・セットを備えています。「編集」メニューに含まれているのは「コピー」オプションのみです。拡張の出力ノードの出力ブラウザーには、2つのタブが表示されます。

- 「テキスト出力」タブには、テキスト出力が表示されます。
- 「グラフ出力」タブには、グラフおよび図表が表示されます。

「拡張の出力ノード」ダイアログ・ボックスの「出力」タブで「画面に出力」の代わりに「ファイルに出力」が選択されている場合、拡張の出力ノードの実行の成功時に出力ブラウザー・ウィンドウは表示されません。

## 拡張の出力ブラウザーの「テキスト出力」タブ

「テキスト出力」タブには、拡張の出力ノードの「シンタックス」タブにある R スクリプトまたは Python for Spark スクリプトが実行された際に生成される、すべてのテキスト出力が表示されます。

注: 拡張の出力スクリプトの実行結果として生成される R または Python for Spark のエラー・メッセージや警告は、拡張の出力ノードの「**コンソール出力**」タブに常に表示されます。

### 拡張の出力ブラウザーの「**グラフ出力**」タブ

「**グラフ出力**」タブには、拡張の出力ノードの「**シンタックス**」タブにある R スクリプトまたは Python for Spark スクリプトが実行された際に生成される、すべてのグラフまたは図表が表示されます。例えば、R スクリプトに R の plot 関数への呼び出しが含まれる場合、その結果のグラフがこのタブに表示されます。

## 拡張モデル・ノード

拡張モデル・ノードを使用すると、R スクリプトまたは Python for Spark スクリプトを実行して、モデルの作成とスコアリングを行うことができます。

### 拡張モデル・ノード - 「**シンタックス**」タブ

シンタックスのタイプ (**R** または **Python for Spark**) を選択します。次に、以下のいずれかのフィールドにカスタム・スクリプト・シンタックスを入力するか、貼り付けます。シンタックスの準備ができたなら、「**実行**」をクリックして、拡張モデル・ノードを実行できます。

#### R シンタックス

「**R モデル作成シンタックス**」。このフィールドに、カスタムの R モデル作成スクリプト構文を入力したり、貼り付けたりできます。

「**R モデル スコアリング シンタックス**」。このフィールドに、カスタムの R モデル・スコアリング・スクリプト構文を入力したり、貼り付けたりできます。

#### Python for Spark シンタックス (Python for Spark syntax)

「**Python モデル作成シンタックス (Python model building syntax)**」。このフィールドに、カスタムの Python モデル作成スクリプト構文を入力したり、貼り付けたりできます。

「**Python モデル スコアリング シンタックス (Python model scoring syntax)**」。このフィールドに、カスタムの Python モデル・スコアリング・スクリプト構文を入力したり、貼り付けたりできます。

Python for Spark について詳しくは、[1 ページ](#)の『Python for Spark』および [2 ページ](#)の『Python for Spark を使用したスクリプト』を参照してください。

### 拡張モデル・ノード - 「**モデルオプション**」タブ

**モデル名:** ターゲットまたは ID フィールド (その指定がない場合はモデル・タイプ) に基づいてモデル名を生成、またはカスタム名を指定することができます。

### 拡張モデル・ノード - 「**コンソール出力**」タブ

「**コンソール出力**」タブには、「**シンタックス**」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、「**シンタックス**」タブの「**R シンタックス**」フィールドにある R スクリプトが実行されたときに R コンソールから受信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトの実行時に生成される R または Python のエラー・メッセージや警告が含まれる場合があります。出力は、主にスクリプトをデバッグするために使用できます。「**コンソール出力**」タブには、「**R シンタックス**」フィールドまたは「**Python シンタックス (Python Syntax)**」フィールドのスクリプトも表示されます。

拡張モデル・スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「**コンソール出力**」タブが上書きされます。出力を編集することはできません。

### 拡張モデル・ノード - 「**テキスト出力**」タブ

「**テキスト出力**」タブは、拡張モデル・ノードのダイアログ・ボックスの「**モデルオプション**」タブの「**R テキスト出力の表示**」チェック・ボックスを選択することによって要求された場合に、拡張モデル・ノードに表示されます。このタブには、テキスト出力のみを表示できます。R モデル作成スクリプトの実行によって生成されるテキスト出力はすべて、このタブに表示されます。モデルに異なる名前を指定せずに、こ

のモデル作成スクリプトを再度実行すると、前の実行による「テキスト出力」タブの内容が上書きされます。テキスト出力を編集することはできません。

スクリプトに R の sink 関数への呼び出しを含めると、この関数後に生成されるすべての出力は、指定したファイルに保存され、「テキスト出力」タブには表示されません。

注：モデル作成スクリプトの実行結果として生成される R または Python for Spark のエラー・メッセージや警告は、拡張モデル・ノードの「コンソール出力」タブに常に表示されます。

## 拡張モデル・ナゲット

拡張モデル・ナゲットは、拡張モデル・ノードの実行後に生成され、モデル・パレットに配置されます。このナゲットには、モデル作成およびモデル・スコアリングを定義する R スクリプトまたは Python for Spark スクリプトが含まれます。デフォルトでは、拡張モデル・ナゲットには、モデル・スコアリングに使用されるスクリプト、データを読み取るためのオプション、および R コンソールまたは Python for Spark からのすべての出力が含まれます。拡張モデル・ナゲットには、オプションで、グラフ、テキスト出力などの他のさまざまな形式のモデル出力も含めることができます。拡張モデル・ナゲットが生成され、ストリーム領域に追加されると、出力ノードをそれに接続できます。次に、この出力ノードが IBM SPSS Modeler ストリーム内で通常の方法で使用されて、データおよびモデルに関する情報の取得、およびさまざまな形式でのデータのエクスポートが実行されます。

このノードを R と共に使用するには、IBM SPSS Modeler - Essentials for R をインストールする必要があります。インストール手順と、互換性に関する情報については、「IBM SPSS Modeler - Essentials for R: インストール手順」を参照してください。また、ご使用のコンピューターに、互換性のあるバージョンの R がインストールされている必要もあります。

### 拡張モデル・ナゲット - 「シンタックス」タブ

「シンタックス」タブは、常に拡張モデル・ナゲットに表示されます。

「R モデル スコアリング シンタックス」。R を使用する場合は、モデル・スコアリングに使用される R スクリプトがこのフィールドに表示されます。デフォルトでは、このフィールドは有効になっていますが、編集はできません。R モデル・スコアリング・スクリプトを編集するには、「編集」をクリックします。

「Python モデル スコアリング シンタックス (Python model scoring syntax)」。Python for Spark を使用する場合は、モデル・スコアリングに使用される Python スクリプトがこのフィールドに表示されます。デフォルトでは、このフィールドは有効になっていますが、編集はできません。Python モデル・スコアリング・スクリプトを編集するには、「編集」をクリックします。

「編集」。「編集」をクリックして、スコアリング・シンタックス・フィールドを編集可能にします。これにより、スコアリング・シンタックス・フィールドに入力することで、モデル・スコアリング・スクリプトを編集できるようになります。例えば、拡張モデル・ナゲットの実行後、モデル・スコアリング・スクリプトにエラーを見つけた場合に、モデル・スコアリング・スクリプトを編集することが考えられます。拡張モデル・ナゲット内でモデル・スコアリング・スクリプトに加えた変更は、拡張モデル・ノードの実行によってモデルを再生成すると失われます。

### 拡張モデル・ナゲット - 「モデル オプション」タブ

「モデル オプション」タブは、常に拡張モデル・ナゲットに表示されます。

「データ読み取りオプション」。これらのオプションは、R にのみ適用され、Python for Spark には適用されません。これらのオプションにより、欠損値、フラグ型フィールド、および日付形式または日付/時刻形式の変数の処理方法を指定できます。

- 「データをバッチで読み込む」。処理するデータ量が多い場合 (例えば、多すぎて R エンジンのメモリーに収まらない場合)、このオプションを使用して、別個に送信および処理できるバッチにデータを分割します。個々のバッチに含めるデータレコードの最大数を指定します。

拡張の変換ノードおよび拡張スコアリング・ナゲットの両方の場合、データは R スクリプトを使用して (バッチで) 渡されます。そのため、Hadoop 環境またはデータベース環境で実行されるモデル スコアリングとプロセス ノード用のスクリプトには、ソートや集計など、データ内の複数の行にまたがる操作や、複数の行を結合する操作を含めることはできません。この制約により、Hadoop 環境や、データベース内マイニングの間に、確実にデータが分割された状態になるよう強制することができます。モデル スコア

リング用のスクリプトを SPSS Modeler Server で実行する場合、この制約は適用されません。拡張の出力ノードおよび拡張モデル・ノードには、この制約はありません。

- 「**フラグ型フィールドの変換**」。フラグ型フィールドの処理方法を指定します。「**文字列を因子に、整数および実数を倍精度に**」および「**論理値(真、偽)**」の2つのオプションがあります。「**論理値(真、偽)**」を選択した場合、フラグ型フィールドの元の値は失われます。例えば、フィールドに「Male」および「Female」という値がある場合、これらの値は「真」および「偽」に変更されます。
- 「**欠損値を R の欠損値 (NA) に変換**」。選択すると、欠損値はすべて、R の NA 値に変換されます。R では、欠損値の識別に値 NA が使用されます。使用する R 関数によっては、データに NA が含まれていた場合の関数の動作を制御するために使用される引数が含まれている場合があります。例えば、関数によって NA を含むレコードを自動的に除外することを選択できる場合があります。このオプションが選択されない場合、すべての欠損値はそのまま R に渡されます。これらの欠損値は R スクリプトの実行時にエラーの原因となる可能性があります。
- **時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換**: 選択すると、日付形式または日付/時刻形式の変数が R の日付/時刻オブジェクトに変換されます。次のいずれかのオプションを選択する必要があります。
  - 「**R POSIXct**」。日付形式または日付/時刻形式の変数が R の POSIXct オブジェクトに変換されます。
  - 「**R POSIXlt (リスト)**」。日付形式または日付/時刻形式の変数が R の POSIXlt オブジェクトに変換されます。

**注:** POSIX 形式は、拡張オプションです。これらのオプションは、ご使用の R スクリプトで、これらの形式を必要とする方法で日付/時刻フィールドを処理するように指定している場合のみ使用してください。POSIX 形式は、時刻形式の変数には適用されません。

データベースに対して拡張モデル・ナゲットを実行する場合、「**フラグ型フィールドの変換**」、「**欠損値を R の欠損値 (NA) に変換**」および「**時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換**」の各コントロールに対して選択したオプションは認識されません。データベースに対してノードを実行する場合、これらのコントロールに対しては、代わりに以下のデフォルト値が使用されます。

- 「**フラグ型フィールドの変換**」は「**文字列を因子に、整数および実数を倍精度に**」に設定されます。
- 「**欠損値を R の欠損値 (NA) に変換**」が選択されます。
- 「**時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換**」は選択されません。

#### 拡張モデル・ナゲット - 「グラフ出力」タブ

「**グラフ出力**」タブは、拡張モデル・ノードのダイアログ・ボックスの「**モデル オプション**」タブの「**R グラフを HTML として表示**」チェック・ボックスを選択することによって要求された場合に、拡張モデル・ナゲットに表示されます。モデル作成 R スクリプトの実行結果として生成されるグラフをこのタブに表示できます。例えば、R スクリプトに R の plot 関数への呼び出しが含まれる場合、その結果のグラフがこのタブに表示されます。モデルに異なる名前を指定せずに、このモデル作成スクリプトを再度実行すると、前の実行による「**グラフ出力**」タブの内容が上書きされます。

#### 拡張モデル・ナゲット - 「テキスト出力」タブ

「**テキスト出力**」タブは、拡張モデル・ノードのダイアログ・ボックスの「**モデル オプション**」タブの「**R テキスト出力の表示**」チェック・ボックスを選択することによって要求された場合に、拡張モデル・ナゲットに表示されます。このタブには、テキスト出力のみを表示できます。拡張モデル・スクリプトの実行によって生成されるテキスト出力はすべて、このタブに表示されます。モデルに異なる名前を指定せずに、この拡張モデル・スクリプトを再度実行すると、前の実行による「**テキスト出力**」タブの内容が上書きされます。テキスト出力を編集することはできません。

**注:**

- スクリプトに R の sink 関数への呼び出しを含めると、この関数後に生成されるすべての出力は、指定したファイルに保存され、「**テキスト出力**」タブには表示されません。
- 拡張モデル・スクリプトの実行結果として生成されるエラー・メッセージや警告は、拡張モデル・ノードの「**コンソール出力**」タブに常に表示されます。



## 拡張モデル・ナゲット - 「コンソール出力」タブ

「コンソール出力」タブは、常に拡張モデル・ナゲットに表示されます。このタブには、「シンタックス」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、拡張モデル・ナゲットの「シンタックス」タブの「R モデルスコアリング シンタックス」フィールドにある R スクリプトが実行されたときに R コンソールから受信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトが実行されたときに生成される R または Python のすべてのエラー・メッセージや警告と、R コンソールからのすべてのテキスト出力が含まれます。出力は、主にスクリプトをデバッグするために使用できます。

モデル・スコアリング・スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「コンソール出力」タブが上書きされます。コンソール出力を編集することはできません。

## 拡張の変換ノード

拡張の変換ノードを使用すると、IBM SPSS Modeler ストリームからデータを取得し、R スクリプトまたは Python for Spark スクリプトを使用して、取得したデータに変換を適用することができます。変更されたデータは、さらなる処理、モデル構築、およびモデル・スコアリングのために、ストリームに戻されます。拡張の変換ノードを使用すると、R または Python for Spark で記述したアルゴリズムを使用して、データを変換することができます。また、ユーザーは、拡張の変換ノードを使用して、特定の問題に対応したデータ変換手法を開発することができます。

このノードを R と共に使用するには、IBM SPSS Modeler - Essentials for R をインストールする必要があります。インストール手順と、互換性に関する情報については、「IBM SPSS Modeler - Essentials for R: インストール手順」を参照してください。また、ご使用のコンピューターに、互換性のあるバージョンの R がインストールされている必要もあります。

### 拡張の変換ノードの「シンタックス」タブ

シンタックスのタイプ (R または Python for Spark) を選択します。詳しくは、以下のセクションを参照してください。シンタックスの準備ができたなら、「実行」をクリックして、拡張の変換ノードを実行できます。

### R シンタックス

「R シンタックス」。データ分析用のカスタムの R スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。

「フラグ型フィールドの変換」。フラグ型フィールドの処理方法を指定します。「文字列を因子に、整数および実数を倍精度に」および「論理値 (真、偽)」の 2 つのオプションがあります。「論理値 (真、偽)」を選択した場合、フラグ型フィールドの元の値は失われます。例えば、フィールドに「Male」および「Female」という値がある場合、これらの値は「真」および「偽」に変更されます。

「欠損値を R の欠損値 (NA) に変換」。選択すると、欠損値はすべて、R の NA 値に変換されます。R では、欠損値の識別に値 NA が使用されます。使用する R 関数によっては、データに NA が含まれていた場合の関数の動作を制御するために使用される引数が含まれている場合があります。例えば、関数によって NA を含むレコードを自動的に除外することを選択できる場合があります。このオプションが選択されない場合、すべての欠損値はそのまま R に渡されます。これらの欠損値は R スクリプトの実行時にエラーの原因となる可能性があります。

「時間帯を考慮した特殊な制御で日時フィールドを R のクラスに変換」。選択すると、日付形式または日付/時刻形式の変数が R の日付/時刻形式に変換されます。次のいずれかのオプションを選択する必要があります。

- 「R POSIXct」。日付形式または日付/時刻形式の変数が R の POSIXct オブジェクトに変換されます。
- 「R POSIXlt (リスト)」。日付形式または日付/時刻形式の変数が R の POSIXlt オブジェクトに変換されます。

注: POSIX 形式は、拡張オプションです。これらのオプションは、ご使用の R スクリプトで、これらの形式を必要とする方法で日付/時刻フィールドを処理するように指定している場合にのみ使用してください。POSIX 形式は、時刻形式の変数には適用されません。

## Python シンタックス (Python Syntax)

「**Python シンタックス (Python Syntax)**」。データ分析用のカスタムの Python スクリプト・シンタックスを、このフィールドに入力するか、貼り付けることができます。Python for Spark について詳しくは、[Python for Spark](#) および [Python for Spark を使用したスクリプト](#)を参照してください。

### 拡張の変換ノード - 「コンソール出力」タブ

「**コンソール出力**」タブには、「シンタックス」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、「シンタックス」タブの「**R シンタックス**」フィールドにある R スクリプトが実行されたときに R コンソールから受信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトの実行時に生成される R または Python のエラー・メッセージや警告が含まれる場合があります。出力は、主にスクリプトをデバッグするために使用できます。「**コンソール出力**」タブには、「**R シンタックス**」フィールドまたは「**Python シンタックス (Python Syntax)**」フィールドのスクリプトも表示されます。

拡張変換スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「**コンソール出力**」タブが上書きされます。出力を編集することはできません。

## 拡張のインポート・ノード

拡張のインポート・ノードを使用すると、R スクリプトまたは Python for Spark スクリプトを実行して、データをインポートできます。

### 拡張のインポート・ノード - 「シンタックス」タブ

シンタックスのタイプ (**R** または **Python for Spark**) を選択します。次に、データをインポートするためのカスタム・スクリプトを入力するか、貼り付けます。シンタックスの準備ができたなら、「**実行**」をクリックして、拡張のインポート・ノードを実行できます。

### 拡張のインポート・ノード - 「コンソール出力」タブ

「**コンソール出力**」タブには、「シンタックス」タブの R スクリプトまたは Python for Spark スクリプトが実行されたときに受信するすべての出力が含まれます (例えば、R スクリプトを使用する場合、「シンタックス」タブの「**R シンタックス**」フィールドにある R スクリプトが実行されたときに R コンソールから受信する出力が表示されます)。この出力には、R スクリプトまたは Python スクリプトの実行時に生成される R または Python のエラー・メッセージや警告が含まれる場合があります。出力は、主にスクリプトをデバッグするために使用できます。「**コンソール出力**」タブには、「**R シンタックス**」フィールドまたは「**Python シンタックス (Python Syntax)**」フィールドのスクリプトも表示されます。

拡張のインポート・スクリプトが実行されるたびに、R コンソールまたは Python for Spark から受信した出力で「**コンソール出力**」タブが上書きされます。出力を編集することはできません。

## フィールドのフィルタリングまたは名前の変更

ストリーム内の任意の場所でフィールドの名前を変更したり、フィールドを除外することができます。例えば、患者 (レコード・レベル・データ) のカリウム値 (フィールド・レベル・データ) を重要視していない場合、「K」 (カリウム値) フィールドを除外できます。個々のフィルター・ノード、または入力ノードか出力ノードの「**フィルター**」タブを使用して実行することができます。どのノードからアクセスしているかに関係なく、機能は同じです。

- 可変長ファイル、固定長ファイル、Statistics ファイル、XML、拡張のインポートなどの入力ノードから、IBM SPSS Modeler にデータを読み込むフィールドの名前を変更したり、フィルタリングを行うことができます。
- フィルター・ノードを使用して、ストリーム中の任意の場所でフィールドの名前を変更したり、フィルタリングすることができます。
- Statistics エクスポート・ノード、Statistics 変換ノード、Statistics モデル・ノードおよび Statistics 出力ノードから、フィールドをフィルタリングしたり、IBM SPSS Statistics の命名規則に従うようにフィールドの名前を変更することができます。
- 上記のノードいずれかの「**フィルター**」タブを使用して、複数の回答セットを定義または編集することができます。



- 最後にフィルター・ノードを使用して、ある入力ノードから別のノードへフィールドをマップすることができます。



## 第2章 拡張機能

拡張機能は、IBM SPSS Modeler の機能を拡張するカスタム コンポーネントです。拡張機能は拡張バンドル (.mpe ファイル) にパッケージされて、IBM SPSS Modeler にインストールされます。拡張はどのユーザーも作成することができ、関連付けられた拡張バンドルを共有することで他のユーザーと共有できます。

拡張を処理するために、以下のユーティリティーが用意されています。

- 23 ページの『[拡張ハブ](#)』。これは、「[拡張機能](#)」 > 「[拡張ハブ](#)」からアクセスでき、GitHub の IBM SPSS Predictive Analytics コレクションから拡張を検索、ダウンロード、およびインストールするためのインターフェースとして機能します。「[拡張ハブ](#)」ダイアログから、コンピューターにインストールされている拡張の詳細を表示したり、インストール済み拡張の更新を取得したり、拡張を削除したりすることもできます。
- 「[拡張](#)」 > 「[ローカル拡張バンドルのインストール](#)」から、ローカル・コンピューターに保存されている拡張バンドルをインストールできます。
- [拡張用のカスタム ダイアログビルダー](#)を使用して、カスタム ノード ダイアログと呼ばれる、ユーザーインターフェイスを含む拡張を作成できます。カスタム ノード ダイアログによって、拡張に関連付けられているタスクを実行する R スクリプトまたは Python for Spark スクリプトが生成されます。生成されるスクリプトは、カスタム ダイアログの設計の一環として設計します。

### 拡張ハブ

「[拡張ハブ](#)」ダイアログで、以下のタスクを実行できます。

- GitHub の IBM SPSS Predictive Analytics コレクションから入手できる拡張を検索する。拡張機能をすぐにインストールすることも、選択した拡張機能をダウンロードして、後でインストールすることもできます。
- 既にコンピューターにインストールされている拡張機能の更新バージョンを取得する。
- 既にコンピューターにインストールされている拡張機能の詳細を表示する。
- コンピューターにインストールされている拡張機能を削除する。

拡張機能をダウンロードまたは削除するには、以下のようになります。

1. メニューから「[拡張](#)」 > 「[拡張ハブ](#)」を選択します。
2. ダウンロードまたは削除する拡張機能を選択し、「**OK**」をクリックします。「**OK**」をクリックすると、「[探索](#)」タブおよび「[インストール済み](#)」タブのすべての選択内容が処理されます。

デフォルトでは、ダウンロード対象として選択した拡張機能がコンピューターにダウンロードされ、インストールされます。「[設定](#)」タブで、選択した拡張機能をインストールせず、指定した場所にダウンロードするように選択することができます。「[拡張](#)」 > 「[ローカル拡張バンドルのインストール](#)」を選択して、後でそれらをインストールできます。Windows の場合、拡張バンドルファイルをダブルクリックすることで、拡張をインストールできます。

#### 重要:

- Windows 7 以降の場合、既存の拡張バンドルの更新バージョンをインストールするには、管理者特権を使用して IBM SPSS Modeler を実行する必要があることがあります。管理者特権を使用して IBM SPSS Modeler を開始するには、IBM SPSS Modeler のアイコンを右クリックし、「[管理者として実行](#)」を選択します。特に、1つ以上の拡張バンドルをインストールできなかったことを示すエラーメッセージを受けた場合は、管理者特権を使用して実行してみてください。
- インターネットへの接続がプロキシ経由である場合、「[拡張](#)」 > 「[拡張ハブ](#)」メニュー オプション経由で拡張ハブを開こうとしたときに、「インターネット接続が検出されなかったために、一部の機能が利用できません」などのエラーが表示される場合があります。これを解決するために、以下のパラメーターを

jvm.cfg ファイル (SPSS Modeler のインストール済み環境の config ディレクトリーにあります) の # JVM オプションに追加します。ファイルを保存し、SPSS Modeler を再始動します。

```
options, "-DproxyHost=proxyIP"  
options, "-DproxyPort=proxyPort"
```

注: 拡張機能をインストールするときに同意するライセンスは、「インストール済み」タブで拡張機能の「詳細情報」をクリックすることで、後からいつでも表示できます。

## 「探索」タブ

「探索」タブには、GitHub の IBM SPSS Predictive Analytics コレクション (<https://ibmpredictiveanalytics.github.io/>) から入手できるすべての拡張が表示されます。「探索」タブから、ダウンロードしてインストールする新しい拡張機能を選択したり、コンピューターに既にインストールされている拡張機能の更新を選択したりすることができます。「探索」タブでは、インターネット接続が必要です。

- 拡張ごとに、最新バージョンの番号と、そのバージョンに関連付けられている日付が表示されます。拡張機能の簡単な要約も示されます。コンピューターに既にインストールされている拡張機能の場合、インストール済みのバージョン番号も表示されます。
- 「詳細情報」をクリックして、拡張機能に関する詳細情報を表示できます。更新を入手できる場合、「詳細情報」に更新に関する情報が表示されます。
- 「前提条件」をクリックして、IBM SPSS Modeler - Integration Plug-in for R が必要かどうかなど、拡張機能を実行するための前提条件を表示できます。更新を入手できる場合、「前提条件」に更新に関する情報が表示されます。

## 詳細化

表示される拡張機能のセットを絞り込むことができます。拡張機能の一般的なカテゴリー、拡張機能が実装される言語、拡張機能を提供した組織のタイプ、または拡張機能の状態によって絞り込みます。「カテゴリ」などの各グループに対して、表示される拡張機能のリストを絞り込むために使用する項目を複数選択できます。また、検索項目で絞り込むこともできます。検索では大文字と小文字は区別されません。また、アスタリスク (\*) は普通の文字として扱われ、ワイルドカード検索を意味しません。

- 表示される拡張機能のリストを絞り込むには、「適用」をクリックします。カーソルが「検索」ボックスにあるときに Enter キーを押すと、「適用」をクリックした場合と同様に処理されます。
- 使用可能なすべての拡張機能が表示されるようにリストをリセットするには、「検索」ボックス内のすべてのテキストを削除し、すべての項目を選択解除して、「適用」をクリックします。

## Integration Plug-In の取得方法

IBM SPSS Modeler - Integration Plug-in for R を取得するには、次の手順に従います。

[https://github.com/IBMPredictiveAnalytics/R\\_Essentials\\_Modeler/releases/](https://github.com/IBMPredictiveAnalytics/R_Essentials_Modeler/releases/) または IBM SPSS Statistics コミュニティ (<https://www.ibm.com/products/spss-statistics/support>) で入手できる IBM SPSS Modeler - Essentials for R をインストールします。IBM SPSS Modeler - Essentials for R には、IBM SPSS Modeler - Integration Plug-in for R が含まれています。Essentials for R には、R プログラミング言語は含まれていません。R バージョンの 3.6 がまだインストールされていない場合は、そのインストールを行ってから、IBM SPSS Modeler - Essentials for R をインストールします。<https://cran.r-project.org/> から入手できます。R 3.6.x をダウンロードして、インストールすることをお勧めします。

注: インターネットにアクセスできないコンピューターに Essentials for R をインストールし、Essentials for R に付属の R スクリプトを使用する予定の場合、それらのスクリプトに必要なすべての R パッケージを入手し、それらのパッケージを R にインストールする必要があります。特定の R スクリプトに必要な R パッケージを判別するには、「拡張ハブ」ダイアログを開き(「拡張」 > 「拡張ハブ」)、「インストール済み」タブをクリックして、目的の拡張の「詳細情報」をクリックします。必須の R パッケージが「拡張の詳細」ダイアログにリストされます。R パッケージは、いずれかの R CRAN ミラー・サイトから入手できます。R CRAN ミラー・サイトには <http://www.r-project.org/> からアクセスします。ご使用の R バージョンと一致するバージョンのパッケージを入手してください。バージョン固有のパッケージは、CRAN ミラーサイトの「Contributed Packages」ページのリンクから入手できます。

## 「インストール済み」タブ

「インストール済み」タブには、コンピュータにインストールされているすべての拡張が表示されます。「インストール済み」タブで、インストールされている拡張に対して GitHub の IBM SPSS Predictive Analytics コレクションから入手できる更新の選択や、拡張の削除を実行できます。インストール済み拡張機能の更新を取得するには、インターネット接続が必要です。

- 拡張機能ごとに、インストール済みバージョン番号が表示されます。インターネット接続を使用できるときは、最新バージョンの番号と、そのバージョンに関連付けられている日付が表示されます。拡張機能の簡単な要約も示されます。
- 「詳細情報」をクリックして、拡張機能に関する詳細情報を表示できます。更新を入手できる場合、「詳細情報」に更新に関する情報が表示されます。
- 「前提条件」をクリックして、IBM SPSS Modeler - Integration Plug-in for R が必要かどうかなど、拡張機能を実行するための前提条件を表示できます。更新を入手できる場合、「前提条件」に更新に関する情報が表示されます。

### 詳細化

表示される拡張機能のセットを絞り込むことができます。拡張機能の一般的なカテゴリー、拡張機能が実装される言語、拡張機能を提供した組織のタイプ、または拡張機能の状態によって絞り込みます。「カテゴリ」などの各グループに対して、表示される拡張機能のリストを絞り込むために使用する項目を複数選択できます。また、検索項目で絞り込むこともできます。検索では大文字と小文字は区別されません。また、アスタリスク (\*) は普通の文字として扱われ、ワイルドカード検索を意味しません。

- 表示される拡張機能のリストを絞り込むには、「適用」をクリックします。カーソルが「検索」ボックスにあるときに Enter キーを押すと、「適用」をクリックした場合と同様に処理されます。
- 使用可能なすべての拡張機能が表示されるようにリストをリセットするには、「検索」ボックス内のすべてのテキストを削除し、すべての項目を選択解除して、「適用」をクリックします。

### 専用拡張

専用拡張は、ご使用のコンピュータにインストールされているが、GitHub の IBM SPSS Predictive Analytics コレクションからは入手できない拡張です。表示される拡張のセットを絞り込む機能、および拡張を実行するための前提条件を表示する機能は、専用拡張には使用できません。

注：インターネット接続なしに拡張ハブを使用するときは、「インストール済み」タブの一部の機能が使用できない場合があります。

## 設定

「設定」タブは、ダウンロード対象として選択された拡張機能がダウンロードされた後にインストールされるか、ダウンロードされてもインストールされないかを指定します。この設定は、新しい拡張機能と、既存の拡張機能への更新に適用されます。拡張機能をダウンロードして組織内の他のユーザーに配布する場合、拡張機能のダウンロードのみ行い、インストールは実行しないように選択できます。拡張機能を実行するための前提条件が準備されておらず、前提条件を用意する計画である場合も、ダウンロードのみを行い、インストールは実行しないように選択できます。

拡張をインストールせずにダウンロードするように選択した場合、「拡張」 > 「ローカル拡張バンドルのインストール」を選択して、後でインストールできます。Windows の場合、拡張バンドルファイルをダブルクリックすることで、拡張をインストールできます。

## 拡張機能の詳細

「拡張の詳細」ダイアログボックスには、拡張機能の作成者が入力した情報が表示されます。作成者は、要約、バージョンなどの必須情報に加えて、作成者のホーム ページなど、関連する場所への URL を入力している場合があります。拡張が拡張ハブからダウンロードされた場合、「ライセンスの表示」をクリックして表示できるライセンスが含まれています。

「カスタム ノード」。カスタム ノード テーブルには、拡張に含まれるカスタム ノード ダイアログがリストされます。

注: カスタム ノード ダイアログが含まれる拡張のインストールで、「カスタム ノード」テーブルにそのノード ダイアログのエントリを表示するには、IBM SPSS Modeler の再始動が必要な場合があります。

「**従属関係**」。「**従属関係**」グループには、拡張機能に含まれるコンポーネントを実行するために必要なアドオンが表示されます。

- 「**Integration Plug-In for R**」。拡張のコンポーネントは、Integration Plug-in for R を必要とする場合があります。
- 「**R パッケージ**」。拡張機能に必要な R パッケージが表示されます。詳しくは、[27 ページの『必須の R パッケージ』](#)のトピックを参照してください。

## ローカルの拡張バンドルのインストール

ローカル・コンピューターに格納されている拡張バンドルをインストールするには、以下を行います。

1. メニューから次の項目を選択します。

「**拡張**」 > 「**ローカル拡張バンドルのインストール...**」

2. 拡張バンドルを選択します。拡張バンドルのファイル タイプは mpe です。

**重要:** Windows 7 以降のバージョンの Windows を使用しているユーザーの場合、既存の拡張バンドルの更新バージョンをインストールするには、管理者特権を使用して IBM SPSS Modeler を実行する必要があることがあります。管理者特権を使用して IBM SPSS Modeler を開始するには、IBM SPSS Modeler のアイコンを右クリックし、「**管理者として実行**」を選択します。特に、1 つ以上の拡張バンドルをインストールできなかったことを示すエラー メッセージを受け取った場合は、管理者特権を使用して実行してみてください。

### 拡張のインストール場所

デフォルトでは、拡張は、ご使用のオペレーティング システムの、一般ユーザーが書き込み可能な場所にインストールされます。

IBM\_SPSS\_MODELER\_EXTENSIONS\_PATH 環境変数を使用してパスを定義することで、このデフォルトの場所をオーバーライドできます。指定する場所は、対象コンピューターに存在する必要があります。IBM\_SPSS\_MODELER\_EXTENSIONS\_PATH を設定した後は、変更を有効にするために IBM SPSS Modeler を再起動する必要があります。

Windows で環境変数を作成するには、コントロール パネルから次の手順を実行します。

#### Windows 7

1. 「**ユーザー・アカウント**」を選択します。
2. 「**環境変数の変更**」を選択します。
3. 「**新規**」をクリックし、「**変数名**」フィールドに環境変数の名前 (IBM\_SPSS\_MODELER\_EXTENSIONS\_PATH など) を入力して、さらに変数値フィールドにパス (複数指定可) を入力します。

#### Windows 8 以降

1. 「**システム**」を選択します。
2. 「**詳細設定**」タブを選択し、「**環境変数**」をクリックします。「**詳細設定**」タブには、「**システムの詳細設定**」からアクセスします。
3. ユーザー環境変数セクションで「**新規**」をクリックし、「**変数名**」フィールドに環境変数の名前 (IBM\_SPSS\_MODELER\_EXTENSIONS\_PATH など) を入力して、さらに変数値フィールドにパス (複数指定可) を入力します。

**重要:** Windows 7 以降のバージョンの Windows を使用しているユーザーの場合、既存の拡張バンドルの更新バージョンをインストールするには、管理者特権を使用して IBM SPSS Modeler を実行する必要があることがあります。管理者特権を使用して IBM SPSS Modeler を開始するには、IBM SPSS Modeler のアイコン



ンを右クリックし、「管理者として実行」を選択します。特に、1つ以上の拡張バンドルをインストールできなかったことを示すエラーメッセージを受け取った場合は、管理者特権を使用して実行してみてください。

## 必須の R パッケージ

インターネットにアクセスできない場合は、インターネットにアクセスできる他のユーザーから、特定の拡張に対する必須の R パッケージで、ご使用のコンピューター上に見つからないものを入手する必要があります。拡張をインストールすると、「拡張の詳細」ダイアログ ボックスから必須の R パッケージのリストを表示できます。詳しくは、25 ページの『[拡張機能の詳細](#)』のトピックを参照してください。パッケージは <http://www.r-project.org/> からダウンロードし、R 内からインストールできます。詳しくは、R に付属の「R のインストールと管理ガイド」を参照してください。

注：UNIX (Linux を含む) ユーザーの場合、パッケージはソース形式でダウンロードされ、コンパイルされます。この場合、適切なツールをコンピューターにインストールする必要があります。詳しくは、「R のインストールと管理ガイド」を参照してください。特に Debian ユーザーは、`apt-get install r-base-dev` から `r-base-dev` パッケージをインストールする必要があります。

## カスタム・ノードの作成と管理

拡張用のカスタム ダイアログ ビルダーは、SPSS Modeler ストリーム内で使用するためのノードを作成します。

拡張用のカスタム・ダイアログ・ビルダーを使用すると、次のことができます。

- R または Apache Spark (Python 経由) で実装されたノードを実行するための、カスタム・ノード・ダイアログを作成する。詳しくは、30 ページの『[スクリプト・テンプレートの作成](#)』を参照してください。
- カスタム・ノード・ダイアログ (別のユーザーが作成したもの) の指定が含まれたファイルを開き、自分の IBM SPSS Modeler インストール環境にそのダイアログを追加する。必要に応じて、独自に変更を加える。
- 他のユーザーが IBM SPSS Modeler インストール環境に追加できるように、カスタム・ノード・ダイアログの指定を保存する。
- カスタム ノードを作成し、Python for Spark スクリプトを記述して、データソースがある任意の場所からデータを読み取り、Apache Spark によってサポートされている任意のデータ形式にデータを書き出します。詳しくは、56 ページの『[Python for Spark を使用したデータのインポートおよびエクスポート](#)』を参照してください。
- カスタム ノードを作成し、R スクリプトを記述して、データソースがある任意の場所からデータを読み取り、R によってサポートされている任意のデータ形式にデータを書き出します。詳しくは、56 ページの『[R を使用したデータのインポートおよびエクスポート](#)』を参照してください。

拡張用のカスタム・ダイアログ・ビルダーでは、拡張内でカスタム・ノード・ダイアログを作成または変更します。拡張用のカスタム・ダイアログ・ビルダーを開くと、空のカスタム・ノード・ダイアログを含む新しい拡張が作成されます。拡張用のカスタム・ダイアログ・ビルダーからカスタム・ノード・ダイアログを保存またはインストールすると、そのダイアログは拡張の一部として保存またはインストールされます。

注：

- 標準の IBM SPSS Modeler ノードに対して、独自のバージョンのノード・ダイアログを作成することはできません。
- スクリプトは、カスタム・ダイアログ・ビルダーで作成されたノード (カスタム・ダイアログ・ビルダー R ノードおよびカスタム・ダイアログ・ビルダー Python ノードを含む) ではサポートされません。

### 拡張用のカスタム・ダイアログ・ビルダーを起動する方法

メニューから、「拡張」 > 「カスタム ノード ダイアログ ビルダー」を選択します。

注：

- Python ノードは Spark 環境に依存します。
- Python スクリプトでは、データが Spark DataFrame 形式で表示されるため、Spark API を使用する必要があります。
- バージョン 17.1 で作成された古いノードは、引き続き IBM SPSS Analytic Server に対してのみ実行されます (データは IBM SPSS Analytic Server 入力ノードから発信され、IBM SPSS Modeler サーバーには抽出されていません)。バージョン 18.0 以降で作成された新しい Python およびカスタム・ダイアログ・ビルダーのノードは IBM SPSS Modeler サーバーに対して実行できます。
- Python をインストールする際に、必ずすべてのユーザーに Python インストール済み環境にアクセスする許可を付与してください。
- Machine Learning Library (MLlib) を使用する場合、NumPy を含む Python のバージョンをインストールする必要があります。その後、Python インストール済み環境を使用するように IBM SPSS Modeler サーバー (または IBM SPSS Modeler クライアントのローカル・サーバー) を構成する必要があります。詳しくは、[2 ページの『Python for Spark を使用したスクリプト』](#)を参照してください。

## カスタム・ダイアログ・ビルダーのレイアウト

### ダイアログ・キャンバス

ダイアログ・キャンバスとは、ノード・ダイアログのレイアウトをデザインする、カスタム・ダイアログ・ビルダーの領域です。

### プロパティ・ペイン

プロパティ・ペインは、ノード・ダイアログを構成するコントロールのプロパティ、およびダイアログ自体のプロパティ (ノード・タイプなど) を指定する カスタム・ダイアログ・ビルダーの領域です。

### ツール・パレット

ツール・パレットは、カスタム・ノード・ダイアログに含めることができるコントロールのセットを提供しています。「表示」メニューから「ツールパレット」を選択して、ツール・パレットの表示と非表示を切り替えることができます。

### スクリプト・テンプレート

スクリプト・テンプレートは、カスタム・ノード・ダイアログによって生成される R スクリプトまたは Spark 用 Python スクリプトを指定します。スクリプト・テンプレート・ペインは、「[新しいウィンドウに移動](#)」をクリックして、別個のウィンドウに移動できます。別個のスクリプト・テンプレート・ウィンドウをカスタム・ダイアログ・ビルダーに戻すには、「[メインウィンドウに復元](#)」をクリックします。

## カスタム・ノード・ダイアログの作成

カスタム・ノード・ダイアログを作成するための基本的な手順は、次のとおりです。

1. ノード・ダイアログの起動時に表示されるタイトルなど、ノード・ダイアログ自体のプロパティと、IBM SPSS Modeler パレット内の新規ノードの場所を指定します。詳しくは、[29 ページの『ダイアログのプロパティ』](#)のトピックを参照してください。
2. ノード・ダイアログおよび任意のサブダイアログを構成する、フィールド・ピッカーおよびチェック・ボックスなどのコントロールを指定します。詳しくは、[32 ページの『コントロール・タイプ』](#)のトピックを参照してください。
3. ノード・ダイアログによって生成される R コードまたは Python for Spark スクリプトを指定する、スクリプト・テンプレートを作成します。詳しくは、[30 ページの『スクリプト・テンプレートの作成』](#)のトピックを参照してください。
4. ノード・ダイアログが含まれている拡張のプロパティを指定します。詳しくは、[50 ページの『拡張プロパティ』](#)のトピックを参照してください。



5. IBM SPSS Modeler にノード ダイアログが含まれている拡張をインストールするか、拡張バンドル (.mpe) ファイルに拡張を保存するか、またはその両方を実行します。詳しくは、[52 ページの『カスタム・ノード・ダイアログの管理』](#)のトピックを参照してください。

作成中のノード・ダイアログをプレビューすることができます。詳しくは、[31 ページの『カスタム・ノード・ダイアログのプレビュー』](#)のトピックを参照してください。

## ダイアログのプロパティ

カスタム・ダイアログ・ビルダーのウィンドウには、ノード・ダイアログのプロパティおよび選択されたユーザー・インターフェース・コントロールのプロパティが表示されます。ダイアログ・プロパティの表示と設定を行うには、キャンバス上のコントロールの外部の領域をクリックします。キャンバスにコントロールがない場合、ダイアログのプロパティは常に表示されます。

「**ダイアログ名**」。「ダイアログ名」プロパティは必須で、ノード・ダイアログに関連する固有の名前を指定します。名前が競合する可能性を最小限に抑えるために、名前に URL などの組織の識別子を接頭辞として使用できます。

「**タイトル**」。「タイトル」プロパティは、ノード・ダイアログ・ボックスのタイトル・バーに表示されるテキストを指定します。

「**ヘルプ・ファイル**」。「ヘルプ・ファイル」プロパティはオプションで、ノード・ダイアログのヘルプ・ファイルへのパスを指定します。これは、ユーザーがダイアログの「ヘルプ」ボタンをクリックすると起動するファイルです。ヘルプ・ファイルは HTML 形式でなければなりません。ノード ダイアログがインストールされるか、保存されるときに、指定したヘルプ ファイルのコピーがノード ダイアログの仕様を含められます。ランタイム・ダイアログの「ヘルプ」ボタンは、関連するヘルプ・ファイルがない場合は表示されません。

- 同じディレクトリにヘルプ ファイルとして存在するヘルプ ファイルのローカライズ版は、ヘルプ ファイルの追加時に、自動的にノード ダイアログに追加されます。ヘルプ ファイルのローカライズ版には、<Help File>\_<language identifier>.htm という名前がついています。詳しくは、[54 ページの『ローカライズされたカスタム・ノード・ダイアログの作成』](#)のトピックを参照してください。
- まず、ノード ダイアログを保存することで、イメージ ファイル、スタイルシートなどのサポート ファイルをノード ダイアログに追加できます。次に、サポート ファイルを手動でノード ダイアログ ファイル (.cfe) に追加します。カスタム ノード ダイアログにアクセスして手動で変更する方法については、トピック [54 ページの『ローカライズされたカスタム・ノード・ダイアログの作成』](#)の『ダイアログの文字列をローカライズするには』というタイトルのセクションを参照してください。

「**スクリプト・タイプ**」。スクリプト・テンプレートの構築に使用できるスクリプトのタイプを指定します。IBM SPSS Modeler で使用できるのは R スクリプトまたは Spark 用 Python スクリプトです。

「**モデルからのスコア**」。モデル作成スクリプトを使用して構築されたモデルをスコアリングに使用するかどうかを指定します。

「**ノード・タイプ**」。ノード・ダイアログのインストール時に作成されるノードのタイプを指定します。

「**パレット**」。ノード・ダイアログのインストール時に、新規に作成されたノードが追加されるパレットを指定します。

「**ノード・アイコン**」。省略記号 (...) ボタンをクリックして、新規に作成されたノードのノード・アイコンとして使用する画像を選択してください。選択する画像は .gif ファイルである必要があります。

## ダイアログ・キャンバスのコントロールのレイアウト

コントロールをツール・パレットからダイアログ・キャンバスにドラッグして、カスタム・ノード・ダイアログに追加します。組み込みのノード ダイアログとの一貫性を確保するために、ダイアログ キャンバスは、4つの機能列に分割されていて、これらの列にコントロールを配置できます。

- 最初の (左端) の列は、主にフィールド・ピッカー・コントロールに使用されます。
- サブダイアログ・ボタンは、右端の列に配置する必要があります (例えば、3列しか使用されない場合は、3番目の列)。また、サブダイアログ・ボタンと同じ列に他のコントロールを配置することはできません。このため、4番目の列にはサブダイアログ・ボタンしか含めることができません。

ノード・ダイアログが IBM SPSS Modeler にインストールされると、ダイアログ・キャンバスには表示されませんが、適切なボタン (例えば、「OK」、「キャンセル」、「適用」、「リセット」、そして該当する場合は「ヘルプ」および「実行」) がダイアログに追加されます。これらのボタンの存在と場所は、自動的に決まります。ただし、ノード・ダイアログに関連付けられたヘルプ・ファイル (「ダイアログ・プロパティ」の「ヘルプ・ファイル」プロパティで指定) がない場合、「ヘルプ」ボタンは表示されません。

コントロールを上または下へドラッグして、列内のコントロールの縦方向の順序を変更できますが、コントロールの正確な位置は自動的に決定されます。ダイアログ自体がサイズ変更されると、ランタイム時にコントロールは適切な方法でサイズ変更されます。フィールド・ピッカーなどのコントロールは自動的に拡張され、それらの下にある使用可能なスペースいっぱいに表示されます。

## スクリプト・テンプレートの作成

スクリプト・テンプレートは、カスタム・ノード・ダイアログで生成される R スクリプトまたは Python for Spark スクリプトを指定します。単一のカスタム・ノード・ダイアログを使用して、順次実行される 1 つ以上の操作を指定できます。

スクリプト・テンプレートは、静的テキストによって構成されている場合があります。静的テキストは、静的テキスト・コントロールとは異なります。静的テキストとは、ノードが実行されると必ず生成される R コードまたは Python for Spark コードです。例えば、ユーザー入力に依存しないコマンド名およびサブコマンドの指定は静的テキストです。スクリプト・テンプレートは、コントロール識別子で構成されている場合もあります。コントロール識別子は、ランタイム時に、関連するカスタム・ノード・ダイアログ・コントロールの値で置換されます。例えば、フィールド・ピッカーで指定されたフィールドのセットは、フィールド・ピッカー・コントロールのコントロール識別子で表されます。

### スクリプト・テンプレートを作成するには

1. ユーザー指定の値に依存しない静的テキストの場合は、例えば、R 構築ノードの「**R モデル作成シンタックス**」フィールドで入力するのと同様に、R スクリプトまたは Python for Spark スクリプトを入力します。
2. コントロールで生成された R スクリプトまたは Python for Spark スクリプトを挿入する場所で %Identifier% の形式のコントロール識別子を追加します。ここで、Identifier は、コントロールの識別子プロパティの値です。
  - コントロール識別子を挿入するには、識別子のテーブルで行を選択して右クリックし、「**スクリプト・テンプレートに追加 (Add to script template)**」を選択します。また、キャンバス上のコントロールを右クリックして「**スクリプト・テンプレートに追加 (Add to script template)**」を選択することによってもコントロール識別子を挿入できます。
  - Ctrl キーを押しながらスペース・バーを押して、使用可能なコントロール識別子のリストから選択することも可能です。リストには、スクリプト自動入力機能で使用できる項目の前にコントロール識別子が表示されます。

識別子ではスペースはすべて有意なので、手動で識別子を入力する場合はスペースも保持してください。

実行時に、チェック・ボックス、チェック・ボックス・グループ、および静的テキスト・コントロールを除くすべてのコントロールでは、各識別子が、関連付けられているコントロールの「**スクリプト**」プロパティの現在の値で置き換えられます。実行時にコントロールが空の場合、スクリプトは生成されません。チェック・ボックスとチェック・ボックス・グループの場合は、識別子は、現在の状態 (コントロールがチェックされているかチェックされていないか) に応じて、関連するコントロールのチェックされた R スクリプト・プロパティまたはチェックされていない R スクリプト・プロパティの現在の値で置き換えられます。詳しくは、32 ページの『[コントロール・タイプ](#)』のトピックを参照してください。

## 例: R スクリプト・テンプレートに実行時の値を含める

この例では、カスタム・ノード・ダイアログによって、ここに示すシグニチャーを指定した R `lm` 関数への呼び出しを使用して、線型回帰モデルの構築とスコアリングを行う R スクリプトが生成され、実行されます。

```
lm(formula,data)
```

- `formula` は、`Na~Age` のような式を指定します。ここで `Na` はモデルの対象フィールドであり、`Age` はモデルの入力フィールドです。
- `data` は、この式で指定されるフィールドの値が含まれるデータ・フレームです。

単一フィールドのフィールド・ピッカー・コントロールがある、カスタム・ノード・ダイアログ(ユーザーが、線型モデルの入力フィールドを選択できる)について検討します。「スクリプト」タブで、モデルを構築する R スクリプトを生成し、実行するスクリプト・テンプレートが入力されます。入力内容は、以下のようになります。

```
modelerModel <- lm(Na~%input%,data=modelerData)
```

- `%input%` は、フィールド・ピッカー・コントロールの識別子プロパティの値です。これは、実行時にコントロールの「スクリプト」プロパティの現在の値に置き換えられます。
- フィールド・ピッカー・コントロールの「スクリプト」プロパティを `%ThisValue%` となるように定義すると、実行時に、プロパティの現在の値がコントロール(フィールド・ピッカーから選択されるフィールド)の値になるように指定されます。

カスタム・ノード・ダイアログのユーザーが、モデルの入力フィールドとして年齢フィールドを選択したとします。すると、ノード・ダイアログによって、以下の R スクリプトが生成されます。

```
modelerModel <- lm(Na~Age,data=modelerData)
```

「スコアスクリプト」タブに、モデルをスコアリングする R スクリプトを生成し、実行するスクリプト・テンプレートが入力されます。入力内容は、以下のようになります。

```
result <- predict(modelerModel,newdata=modelerData)
var1 <-c(fieldName="predicted",
fieldLabel="",fieldStorage="real",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)
```

この R スクリプトはユーザー指定の値には依存していません。このスクリプトが依存しているのは、モデル構築 R スクリプトを使用して構築されたモデルのみです。したがって、モデル・スコアリング R スクリプトは、R 構築ノードの「R モデル・スコアリング・シンタックス」フィールドと同じように入力されます。

## カスタム・ノード・ダイアログのプレビュー

カスタム・ダイアログ・ビルダーで現在開いているノード・ダイアログをプレビューできます。ダイアログは、IBM SPSS Modeler 内のノードから実行する場合と同様に表示され、機能します。

- フィールド・ピッカーには、ダミー・フィールドが設定されています。
- 「OK」ボタンでは、プレビューを閉じます。
- ヘルプ・ファイルを指定した場合、「ヘルプ」ボタンが有効になり、指定したファイルが開きます。ヘルプ・ファイルを指定しなかった場合、「ヘルプ」ボタンはプレビュー時に無効となり、実際のダイアログが実行される際に非表示となります。

カスタム・ノード・ダイアログをプレビューするには、カスタム・ダイアログ・ビルダーのメニューから、「ファイル」>「ダイアログをプレビュー」を選択します。

## コントロール・タイプ

ツール・パレットには、カスタム・ノード・ダイアログで必要となる可能性がある、すべての標準コントロールが用意されています。

- **フィールド・ピッカー:** アクティブなデータ・セットのすべてのフィールドのリスト。詳しくは、[32 ページの『フィールド ピッカー』](#)のトピックを参照してください。
- **チェック・ボックス:** 単一のチェック・ボックス。詳しくは、[34 ページの『チェック・ボックス』](#)のトピックを参照してください。
- **コンボ・ボックス:** ドロップダウン・リストを作成するコンボ・ボックス。詳しくは、[34 ページの『コンボボックス』](#)のトピックを参照してください。
- **リスト・ボックス:** 単一選択リストまたは複数選択リストを作成するリスト・ボックス。詳しくは、[34 ページの『コンボボックス』](#)のトピックを参照してください。
- **テキスト・コントロール:** 任意のテキストを入力として受け入れるテキスト・ボックス。詳しくは、[37 ページの『テキスト・コントロール』](#)のトピックを参照してください。
- **数値コントロール:** 入力が数値に制限されているテキスト・ボックス。詳しくは、[38 ページの『数値コントロール』](#)のトピックを参照してください。
- **日付コントロール:** 日時の値を指定するためのスピナー・コントロール。日付、時刻、および日時が含まれています。詳しくは、[39 ページの『日付コントロール』](#)のトピックを参照してください。
- **保護テキスト:** アスタリスクでユーザーによる入力値がマスクされるテキスト・ボックス。詳しくは、[40 ページの『保護テキスト』](#)のトピックを参照してください。
- **静的テキスト・コントロール:** 静的テキストを表示するコントロール。詳しくは、[41 ページの『静的テキスト・コントロール』](#)のトピックを参照してください。
- **カラー・ピッカー:** 色を指定し、関連付けられた RGB 値を生成するためのコントロール。詳しくは、[41 ページの『カラー・ピッカー』](#)のトピックを参照してください。
- **テーブル・コントロール:** 固定数の列と、ランタイム時に追加される可変数の行が含まれたテーブル。詳しくは、[42 ページの『テーブル・コントロール』](#)のトピックを参照してください。
- **項目グループ:** チェック・ボックスのセットなど、コントロールのセットをグループ化するコンテナ。詳しくは、[44 ページの『項目グループ』](#)のトピックを参照してください。
- **ラジオ・グループ:** ラジオ・ボタンのグループ。詳しくは、[44 ページの『ラジオ・グループ』](#)のトピックを参照してください。
- **チェック・ボックス・グループ:** 単一のチェック・ボックスにより、グループとして有効化または無効化されるコントロールのセットのコンテナ。詳しくは、[46 ページの『チェック・ボックス・グループ』](#)のトピックを参照してください。
- **ファイル・ブラウザー:** ファイルを開くか、または保存するファイル・システムを参照するコントロール。詳しくは、[47 ページの『ファイル・ブラウザー』](#)のトピックを参照してください。
- **タブ:** 単一のタブ。詳しくは、[48 ページの『タブ』](#)のトピックを参照してください。
- **サブダイアログ・ボタン:** サブダイアログを起動するボタン。詳しくは、[48 ページの『サブダイアログ・ボタン』](#)のトピックを参照してください。

### フィールド ピッカー

フィールド・ピッカー・コントロールは、ノード・ダイアログのエンド・ユーザーが使用できるフィールドのリストを表示します。アクティブなデータ・セットのすべてのフィールドを表示する(デフォルト)か、または、スケールの測定レベルを持つ数値フィールドなど、タイプや測定レベルに基づいてリストをフィルタリングすることができます。また、他のフィールドピッカーを現在のフィールドピッカーフィールドのソースとして指定することもできます。フィールド・ピッカー・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。

「タイトル」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n`を使用して改行を指定します。

「**表題位置**」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。このプロパティは、ピッカー・タイプが単一フィールドを選択するように設定されている場合にのみ適用されます。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。指定したテキストは、コントロールのタイトル領域にカーソルを移動したときに表示されます。リストされたフィールドのいずれか 1 つの上にカーソルを移動すると、フィールド名とラベルが表示されます。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「**ピッカー・タイプ**」。カスタム・ノード・ダイアログのフィールド・ピッカーを使用して、フィールド・リストから単一のフィールドを選択できるのかまたは複数のフィールドを選択できるのかを指定します。

「**区切り記号の種類**」。生成されたスクリプト内の選択したフィールド間の区切り文字を指定します。使用できる区切り文字は、ブランク、コンマ、および正符号 (+) です。また、任意の単一文字を入力して、区切り文字として使用することもできます。

「**最小フィールド数**」。該当する場合に、コントロールに指定する必要があるフィールドの最小数。

「**最大フィールド数**」。該当する場合に、コントロールに指定できるフィールドの最大数。

「**実行に必要**」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「OK」ボタンをクリックすると、エラーが生成されます。「False」が指定された場合、このコントロールの値は「OK」ボタンの状態に影響を与えません。

「**変数フィルター**」。コントロールに表示されるフィールドのセットをフィルタリングすることができます。フィールドの型および測定レベルでフィルタリングでき、また、フィールドのリストに多重回答セットが含まれるように指定できます。省略符号 (...) ボタンをクリックして、「フィルタリング」ダイアログを開きます。キャンバス上のフィールド・ピッカー・コントロールをダブルクリックして、「フィルター」ダイアログを開くこともできます。詳しくは、[34 ページの『フィールド・リストのフィルタリング』](#)のトピックを参照してください。

「**フィールド ソース**」。別のフィールド・ピッカーが現在のフィールド・ピッカー・フィールドのソースであることを指定します。フィールド・ソース・プロパティが設定されていない場合、フィールドのソースはアクティブなデータ・セットになります。省略符号 (...) ボタンをクリックし、ダイアログ・ボックスを開いて、フィールド・ソースを指定します。

「**スクリプト**」。ランタイムで、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入することができる・スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 `%%ThisValue%%` は、コントロールの実行時の値 (フィールドのリスト) を指定します。これがデフォルトです。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されません。

## フィールド・ピッカーのフィールド・ソースの指定

「**フィールド ソース**」ダイアログ・ボックスは、フィールド・ピッカーに表示されるフィールドのソースを指定します。ソースは他のフィールド・ピッカーにすることができます。選択したコントロールにあるフィールドを表示するように選択することも、アクティブなデータ・セットから、選択したコントロールにないフィールドを表示するように選択することもできます。



## フィールド・リストのフィルタリング

「フィルター」ダイアログ・ボックスはフィールド・ピッカー・コントロールに関連付けられており、リストに表示できるアクティブなデータ・セットからフィールドのタイプをフィルタリングすることができます。また、アクティブなデータ・セットに関連付けられている多重回答セット応答を含めるかどうかも指定できます。数値型フィールドには、日付/時刻形式以外のあらゆる数値書式が含まれます。

## チェック・ボックス

チェック・ボックス・コントロールは、チェックされた状態とチェックされていない状態に対して、それぞれ異なる R スクリプトまたは Python for Spark スクリプトを生成および実行できる、単純なチェック・ボックスです。チェック・ボックス・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。

「タイトル」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「デフォルト値」。チェック・ボックスのデフォルトの状態 (チェックされた状態、またはチェックされていない状態)。

「チェックされた/チェックされていない」「スクリプト」。コントロールがチェックされたとき、およびコントロールのチェックが外されたときに、生成および実行される R スクリプトまたは Spark 用 Python スクリプトを指定します。スクリプト・テンプレートにスクリプトを含めるには、「識別子」プロパティの値を使用します。生成されたスクリプトは、「チェックされたスクリプト」プロパティと、「チェックされていないスクリプト」プロパティのどちらから生成されたかにかかわらず、識別子の指定した位置に挿入されます。例えば、識別子が `checkbox1` である場合、ランタイム時にスクリプト・テンプレートの `%checkbox1%` のインスタンスが、ボックスにチェック・マークが付けられている場合は「チェックされたスクリプト」プロパティの値に、ボックスにチェック・マークが付けられていない場合は「チェックされていないスクリプト」プロパティの値に置き換えられます。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## コンボボックス

コンボ・ボックス・コントロールを使用すると、選択したリスト項目に固有の R スクリプトまたは Spark 用 Python スクリプトを生成および実行できるドロップダウン・リストを作成できます。コンボ・ボックスでは、項目は 1 つしか選択できません。コンボ・ボックス・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。これはスクリプト・テンプレートのコントロールを参照する際に使用する識別子です。

「タイトル」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「表題位置」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「リスト項目」。省略符号 (...) ボタンをクリックして「項目プロパティの一覧表示」ダイアログ・ボックスを開き、コントロールのリスト項目を指定できます。キャンバス上のコンボボックスコントロールをダブルクリックすることで、「項目プロパティの一覧表示」ダイアログを開くこともできます。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「編集可能」。コンボ・ボックス・コントロールが編集可能であるかどうかを指定します。コントロールが編集可能な場合、ランタイム時にカスタム値を入力できます。

「スクリプト」。実行時にこのコントロールで生成され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 値 `%%ThisValue%%` はコントロールの実行時の値を指定し、デフォルトとなります。リスト項目が手動で定義された場合、ランタイム値は、選択したリスト項目の「スクリプト」プロパティの値となります。リスト項目がターゲット・リスト・コントロールを基準とする場合、実行時の値は、選択したリスト項目の値となります。複数選択のリスト・ボックス・コントロールでは、実行時の値は、空白で区切られた、選択した項目のリストとなります。詳しくは、35 ページの『コンボ・ボックスとリスト・ボックスのリスト項目の指定』のトピックを参照してください。
- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。

「引用符の処理」。引用符付き文字列の一部としてスクリプトプロパティに `%%ThisValue%%` が含まれている場合、`%%ThisValue%%` の実行時の値に対する引用符の処理を指定します。このコンテキストでは、引用符付き文字列は単一引用符または二重引用符で囲まれた文字列になります。引用符の処理は、`%ThisValue%` を囲む引用符と同じタイプの引用符にのみ適用されます。以下に示すタイプの引用符の処理を使用することができます。

#### Python

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、コンボ・ボックスのランタイム値が `Combo box's value` である場合、生成されるスクリプトは `'Combo box¥'s value'` になります。`%ThisValue%` が三重引用符で囲まれている場合、引用符の処理は行われないので注意してください。

#### R

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、コンボ・ボックスのランタイム値が `Combo box's value` である場合、生成されるスクリプトは `'Combo box¥'s value'` になります。

#### なし

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、変更されずに保持されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

### コンボ・ボックスとリスト・ボックスのリスト項目の指定

「リスト項目のプロパティ」。ダイアログ・ボックスを使用して、コンボ・ボックス・コントロールまたはリスト・ボックス・コントロールのリスト項目を指定できます。

「手動で定義された値」。各リスト項目を明示的に指定できます。

- 「識別子」。リスト項目の固有の識別子。
- 「名前」。この項目に対してリスト内に表示される名前。名前は必須フィールドです。

- 「**デフォルト**」。コンボ・ボックスの場合は、このリスト項目がコンボ・ボックス内に表示されるデフォルト項目になるかどうかを指定します。リスト・ボックスの場合は、このリスト項目がデフォルトで選択されるかどうかを指定します。
- 「**スクリプト**」。リスト項目が選択されたときに作成される R スクリプトまたは Python for Spark スクリプトを指定します。
- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。

**注:** 既存リストの下部にあるブランク行に新しいリスト項目を追加できます。識別子以外のプロパティのいずれかを入力すると、固有の識別子が生成されます。これは、そのまま使用することもできれば、変更することも可能です。リスト項目は、ボタンの「識別子」セルをクリックしてから delete キーを押すことにより削除できます。

## リスト・ボックス

リスト・ボックス・コントロールを使用すると、1 つまたは複数の項目を選択でき、選択した項目に固有の R スクリプトまたは Python for Spark スクリプトを生成する項目リストを表示できます。リスト・ボックス・コントロールには、次のプロパティがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートのコントロールを参照する際に使用する識別子です。

「**タイトル**」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、¥n を使用して改行を指定します。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「**リスト項目**」。省略符号 (...) ボタンをクリックして「項目プロパティの一覧表示」ダイアログ・ボックスを開き、コントロールのリスト項目を指定できます。キャンバス上のリスト・ボックス・コントロールをダブルクリックして、「項目プロパティの一覧表示」ダイアログを開くこともできます。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「**リストボックスタイプ**」。リスト・ボックスが単一選択だけをサポートするか、または複数選択をサポートするかを指定します。項目がチェック・ボックスのリストとして表示されるように指定することもできます。

「**区切り記号の種類**」。生成されたスクリプト内の選択したリスト項目間の区切り文字を指定します。使用できる区切り文字は、ブランク、コンマ、および正符号 (+) です。また、任意の単一文字を入力して、区切り文字として使用することもできます。

「**最小選択数**」。該当する場合に、コントロールで選択する必要のある項目の最小数。

「**最大選択数**」。該当する場合に、コントロールで選択できる項目の最大数。

「**スクリプト**」。実行時にこのコントロールで生成され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 値 `%%ThisValue%%` はコントロールの実行時の値を指定し、デフォルトとなります。リスト項目が手動で定義された場合、ランタイム値は、選択したリスト項目の「スクリプト」プロパティの値となります。リスト項目がターゲット・リスト・コントロールを基準とする場合、実行時の値は、選択したリスト項目の値となります。複数選択リスト・ボックス・コントロールの場合、実行時の値は指定した区切り記号の種類で区切られた (デフォルトはブランク区切り)、選択した項目のリストになります。詳しくは、[35 ページの『コンボ・ボックスとリスト・ボックスのリスト項目の指定』](#)のトピックを参照してください。
- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。



「引用符の処理」。引用符付き文字列の一部としてスクリプトプロパティに `%%ThisValue%%` が含まれている場合、`%%ThisValue%%` の実行時の値に対する引用符の処理を指定します。このコンテキストでは、引用符付き文字列は単一引用符または二重引用符で囲まれた文字列になります。引用符の処理は、`%ThisValue%` を囲む引用符と同じタイプの引用符にのみ適用されます。以下に示すタイプの引用符の処理を使用することができます。

#### Python

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` で、選択したリスト項目の値が `List item's value` である場合、生成されるスクリプトは `'List item¥'s value'` になります。`%ThisValue%` が三重引用符で囲まれている場合、引用符の処理は行われないので注意してください。

#### R

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` で、選択したリスト項目の値が `List item's value` である場合、生成されるスクリプトは `'List item¥'s value'` になります。

#### なし

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、変更されずに保持されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

### テキスト・コントロール

テキスト・コントロールは、任意の入力を受け入れることができる単純なテキスト・ボックスで、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。これはスクリプト・テンプレートのコントロールを参照する際に使用する識別子です。

タイトル。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「表題位置」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、`Alt` キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「テキストの内容」。内容が任意であるか、IBM SPSS Modeler フィールド名のルールに準拠する文字列がテキストボックスに含まれている必要があるかを指定します。

「デフォルト値」。テキスト・ボックスのデフォルト・コンテンツ。

幅: コントロールのテキスト領域の幅を文字数で指定します。正の整数で指定する必要があります。この値を空にすると、テキスト領域の幅が自動的に決定されます。

「実行に必要」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「OK」ボタンをクリックすると、エラーが生成されます。「False」が指定された場合、このコントロールの値は「OK」ボタンの状態に影響を与えません。デフォルトは **False** です。

「スクリプト」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 `%%ThisValue%%` は、コントロールの実行時の値 (テキスト・ボックスの内容) を指定します。これはデフォルトです。
- スクリプト・プロパティに `%%ThisValue%%` が含まれ、テキスト・ボックスのランタイム値が空である場合、テキスト・ボックス・コントロールはスクリプトを生成しません。

「引用符の処理」。引用符付き文字列の一部としてスクリプトプロパティに `%%ThisValue%%` が含まれている場合、`%%ThisValue%%` の実行時の値に対する引用符の処理を指定します。このコンテキストでは、引用符付き文字列は単一引用符または二重引用符で囲まれた文字列になります。引用符の処理は、`%ThisValue%` を囲む引用符と同じタイプの引用符にのみ適用されます。以下に示すタイプの引用符の処理を使用することができます。

### Python

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、テキスト・コントロールのランタイム値が `Text box's value` である場合、生成されるスクリプトは `'Text box¥'s value'` になります。`%%ThisValue%%` が三重引用符で囲まれている場合、引用符の処理は行われません。

### R

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、テキスト・コントロールのランタイム値が `Text box's value` である場合、生成されるスクリプトは `'Text box¥'s value'` になります。

### なし

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、変更されずに保持されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## 数値コントロール

数値コントロールは、数値を入力するためのテキスト・ボックスで、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。これはスクリプト・テンプレートのコントロールを参照する際に使用する識別子です。

「タイトル」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「表題位置」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「数値の型」。入力内容に関する制限を指定します。「実数」は、数値であるという点を除いて、入力する値に制限がないことを指定します。「整数」は、値が整数でなければならないことを指定します。

「スピン入力 (Spin Input)」。コントロールをスピナーとして表示するかどうかを指定します。デフォルトは False です。

「増分 (Increment)」。コントロールをスピナーとして表示する場合の増分。

「**デフォルト値**」。デフォルト値(ある場合)。

「**最小値**」。許可される最小値(ある場合)。

「**最大値**」。許可される最大値(ある場合)。

「**幅**」。コントロールのテキスト領域の幅を文字数で指定します。正の整数で指定する必要があります。この値を空にすると、テキスト領域の幅が自動的に決定されます。

「**実行に必要**」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「**OK**」ボタンをクリックすると、エラーが生成されます。「**False**」が指定された場合、このコントロールの値は「**OK**」ボタンの状態に影響を与えません。デフォルトは **False** です。

「**スクリプト**」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 `%%ThisValue%%` は、コントロールの実行時の値(数値)を指定します。これはデフォルトです。
- 「スクリプト」プロパティに `%%ThisValue%%` が含まれ、数値コントロールのランタイム時の値が空の場合、数値コントロールはスクリプトを生成しません。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## 日付コントロール

日付コントロールは、日付/時刻値を指定するためのスピナー・コントロールであり、これには日付、時刻、および日時が含まれます。日付コントロールには、次のプロパティがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートのコントロールを参照する際に使用する識別子です。

**タイトル**。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「**表題位置**」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

**型**: コントロールが日付、時刻、または日時のどの値であるかを指定します。

### 日付

コントロールによって、`yyyy-mm-dd` の形式のカレンダー日付が指定されます。デフォルトの実行時の値は、「デフォルト値」プロパティによって指定されます。

### 時間

コントロールによって、`hh:mm:ss` の形式で時刻が指定されます。デフォルトの実行時の値は、現在の時刻です。

### 日時

コントロールによって、`yyyy-mm-dd hh:mm:ss` の形式で日時と時刻が指定されます。デフォルトの実行時の値は、現在の日付と時刻です。

「**デフォルト値**」。タイプが日付である場合の、コントロールのデフォルト・実行時の値。現在の日付または特定の日付を表示するように指定できます。

「**スクリプト**」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 %%ThisValue%% はコントロールの実行時の値を指定します。これはデフォルトです。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されません。

注: 日付コントロールは、リリース 18 より前の IBM SPSS Modeler のリリースではサポートされません。

## 保護テキスト

保護テキスト・コントロールは、アスタリスクでユーザーによる入力値がマスクされるテキスト・ボックスです。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

**タイトル**。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、 $\backslash n$  を使用して改行を指定します。

「**表題位置**」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

**幅**: コントロールのテキスト領域の幅を文字数で指定します。正の整数で指定する必要があります。この値を空にすると、テキスト領域の幅が自動的に決定されます。

「**実行に必要**」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「OK」ボタンをクリックすると、エラーが生成されます。「False」が指定された場合、このコントロールの値は「OK」ボタンの状態に影響を与えません。デフォルトは **False** です。

「**スクリプト**」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 %%ThisValue%% は、コントロールの実行時の値 (テキスト・ボックスの内容) を指定します。これはデフォルトです。
- スクリプト・プロパティに %%ThisValue%% が含まれており、保護テキスト・コントロールのランタイム値が空である場合、保護テキスト・コントロールは R スクリプトまたは Spark 用 Python スクリプトを生成しません。

「**引用符の処理**」。引用符付き文字列の一部としてスクリプト プロパティに %%ThisValue%% が含まれている場合、%%ThisValue%% の実行時の値に対する引用符の処理を指定します。このコンテキストでは、引用符付き文字列は単一引用符または二重引用符で囲まれた文字列になります。引用符の処理は、Encrypt passed value=False の場合にのみ、%%ThisValue%% を囲む引用符と同じタイプの引用符のみに適用されます。以下に示すタイプの引用符の処理を使用することができます。

## Python

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、コントロールのランタイム値が `Secured Text's value` である場合、生成されるスクリプトは `'Secured Text¥'s value'` になります。`%%ThisValue%%` が三重引用符で囲まれている場合、引用符の処理は行われません。

## R

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、円記号 (¥) でエスケープされます。例えば、スクリプト・プロパティが `'%%ThisValue%%'` であり、コントロールのランタイム値が `Secured Text's value` である場合、生成されるスクリプトは `'Secured Text¥'s value'` になります。

## なし

前後の引用符と一致する `%%ThisValue%%` の実行時の値の引用符は、変更されずに保持されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

注: 保護テキスト コントロールは、リリース 18 より前の IBM SPSS Modeler のリリースではサポートされません。

## 静的テキスト・コントロール

静的テキスト・コントロールを使用すると、ノード・ダイアログにテキストのブロックを追加できます。静的テキスト・コントロールのプロパティは次のとおりです。

「識別子」 コントロールの固有の識別子。

「タイトル」。テキスト ブロックの内容。複数行にわたるコンテンツを入力する場合は、`¥n` を使用して改行を指定します。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## カラー・ピッカー

カラー・ピッカー・コントロールは、色を指定し、関連付けられた RGB 値を生成するためのユーザー・インターフェースです。カラー・ピッカー・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

タイトル。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「表題位置」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニモニック・キーを押すとアクティブ化されます。

「スクリプト」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 %%ThisValue%% は、コントロールの実行時の値 (選択した色の RGB 値) を指定します。RGB 値は、R 値、G 値、B 値の順番で、ブランクで区切られた整数のリストとして表示されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

注: カラー ピッカー コントロールは、リリース 18 より前の IBM SPSS Modeler のリリースではサポートされません。

## テーブル・コントロール

テーブル・コントロールは、固定数の列と、ランタイム時に追加される可変数の行が含まれたテーブルを作成します。テーブル・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

「タイトル」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、¥n を使用して改行を指定します。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すと アクティブ化されます。

「ボタンの再配列」。上に移動ボタンと下に移動ボタンがテーブルに追加されるかどうかを指定します。これらのボタンは、テーブルの行を再配列するために、ランタイム時に使用されます。

「テーブル列」。省略符号 (...) ボタンをクリックし、「テーブル列」ダイアログ・ボックスを開きます。ここで、テーブルの列を指定します。

「最小行数」。テーブルに入れる必要のある行の最小数。

「最大行数」。テーブルに入れることができる行の最大数。

「実行に必要」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「OK」ボタンをクリックすると、エラーが生成されます。「False」が指定された場合、このコントロールの値は「OK」ボタンの状態に影響を与えません。

「スクリプト」。実行時にこのコントロールで生成され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 値 %%ThisValue%% はコントロールの実行時の値を指定し、デフォルトとなります。ランタイム値は、一番左の列から開始して、テーブルの各列によって生成されるスクリプトの、空白で区切られたリストです。スクリプト・プロパティに %%ThisValue%% が含まれており、スクリプトを生成する列がない場合、テーブルは全体的にスクリプトを生成しません。
- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

注: テーブル コントロールは、リリース 18 より前の IBM SPSS Modeler のリリースではサポートされません。

### テーブル・コントロールの列の指定

「テーブル列」ダイアログ・ボックスは、テーブル・コントロールの列のプロパティを指定します。

「識別子」列の固有の識別子。

「列名」。テーブルに表示されるとき列の名前。

「内容」。列のデータのタイプを指定します。「実数」という値は、数値であるという点を除いて、入力する値に制限がないことを指定します。「整数」という値は、値が整数でなければならないことを指定します。「いずれか」という値は、入力した値に制限がないことを指定します。「変数名」という値は、IBM SPSS Statistics の有効な変数名に関して値が要件を満たす必要があることを指定します。

「デフォルト値」。ランタイム時に新しい行がテーブルに追加されるとき、この列のデフォルト値(ある場合)。

「区切り記号の種類」。生成されたスクリプト内にある列の値の間の区切り文字を指定します。使用できる区切り文字は、ブランク、コンマ、および正符号(+)です。また、任意の単一文字を入力して、区切り文字として使用することもできます。

「引用符付き」。生成されたスクリプト内で、列内の各値が二重引用符で囲まれるかどうかを指定します。

「引用符の処理」。引用符付きプロパティが TRUE の場合に、列のセル項目での引用符の処理を指定します。引用符の処理は、セル値の二重引用符にのみ適用されます。以下に示すタイプの引用符の処理を使用することができます。

#### Python

セル値の二重引用符が円記号(¥)でエスケープされます。例えば、セル値が This "quoted" value である場合、生成されるスクリプトは "This ¥"quoted¥" value" になります。

#### R

セル値の二重引用符が円記号(¥)でエスケープされます。例えば、セル値が This "quoted" value である場合、生成されるスクリプトは "This ¥"quoted¥" value" になります。

#### なし

セル値の二重引用符は変更されずに保持されます。

「幅(文字)(Width(chars))」。文字内の列の幅を指定します。正の整数で指定する必要があります。

「スクリプト」。ランタイム時にこの列によって生成された R スクリプトまたは Spark 用 Python スクリプトを指定します。テーブル用に生成されるスクリプトは全体的に、テーブルの一番左の列から開始して各列によって生成されるスクリプトの、空白で区切られたリストです。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号(...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 %%ThisValue%% は、列の実行時の値(指定した区切り記号で区切られた、列内の値のリスト)を指定します。
- 列のスクリプト・プロパティに %%ThisValue%% が含まれ、列のランタイム時の値が空の場合、列はスクリプトを生成しません。

注: 「テーブル列」ダイアログの既存リストの下部にあるブランク行に、新しいテーブル列の行を追加できます。識別子以外のプロパティのいずれかを入力すると、固有の識別子が生成されます。これは、そのまま使用することもできれば、変更することも可能です。テーブル列の識別子セルをクリックして、Delete キーを押すことでテーブル列を削除できます。

### コントロールへのリンク

テーブル コントロールをフィールド ピッカー コントロールにリンクできます。テーブル コントロールをフィールド ピッカーにリンクすると、テーブルにはフィールド ピッカー内の各フィールドに対して 1 行が作成されます。フィールド・ピッカーにフィールドを追加して、テーブルに行を追加します。フィールド・

ピッカーからフィールドを削除して、テーブルから行を削除します。リンクされたテーブル コントロールは、例えば、フィールド ピッカーで選択したフィールドのプロパティを指定するために使用できます。

リンクを有効にするには、内容プロパティの変数名の列がテーブルにあり、キャンバスに少なくとも1つのフィールド ピッカー コントロールが存在する必要があります。

テーブル コントロールをフィールド ピッカーにリンクするには、「テーブル列」ダイアログ ボックス処理の「リンク先コントロール」グループで、「使用可能なコントロール」のリストから「フィールド ピッカー」を指定します。次に、リンクを定義する、「**リンクされた列**」と呼ばれるテーブル列を選択します。テーブルがレンダリングされると、「リンクされた列」には、フィールド ピッカーの現在のフィールドが表示されます。リンクできるのは、複数フィールドのフィールド ピッカーのみです。

## 項目グループ

項目グループ・コントロールは他のコントロールのコンテナで、複数のコントロールから生成されたスクリプトをグループ化および制御できます。例えば、サブコマンドのオプション設定を指定するチェック・ボックスのセットがあるが、少なくとも1つのボックスにチェック・マークが付いている場合にサブコマンドのスクリプトのみを生成するとします。これは、項目グループ・コントロールをチェック・ボックス・コントロールのコンテナとして使用することにより実現されます。項目グループに含めることができるコントロールのタイプは、フィールド・ピッカー、チェック・ボックス、コンボ・ボックス、リスト・ボックス、テキスト・コントロール、数値コントロール、静的テキスト、ラジオ・グループ、およびフィールド・ブラウザーです。項目グループ・コントロールには、次のプロパティがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

「**タイトル**」。グループのオプションのタイトル。複数行にわたるタイトルの場合、 $\backslash n$  を使用して改行を指定します。

「**スクリプト**」。実行時に、このコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 項目グループに含まれている任意のコントロールの識別子を含めることができます。実行時に、識別子はコントロールで生成された R スクリプトまたは Python スクリプトに置き換えられます。
- 値 `%%ThisValue%%` は、項目グループの各コントロールで生成された R スクリプトまたは Python スクリプトのリストを生成します。このリストは空白で区切られ、上から下へ、グループ内に表示される順に並べられています。これはデフォルトです。「スクリプト」プロパティに `%%ThisValue%%` が含まれ、項目グループ内のどのコントロールによってもスクリプトが生成されない場合、項目グループは全体として何のスクリプトも生成しません。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## ラジオ・グループ

ラジオ・グループ・コントロールはラジオ・ボタンのセットのコンテナで、ネストされたコントロールのセットを含めることができます。ラジオ・グループ・コントロールには、次のプロパティがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

「**タイトル**」。グループのオプションのタイトル。複数行にわたるタイトルの場合、 $\backslash n$  を使用して改行を指定します。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。



「ラジオ ボタン」。省略符号 (...) ボタンをクリックして「ラジオ グループ プロパティ」ダイアログ・ボックスを開きます。ここで、ラジオ・ボタンのプロパティを指定し、ボタンをグループに追加したりグループから削除したりできます。特定のラジオ・ボタンのコントロールをネストする機能がラジオ・ボタンのプロパティにあり、「ラジオ・グループのプロパティ」ダイアログ・ボックスで設定されます。キャンバス上のラジオ・グループ・コントロールをダブルクリックして、「ラジオ・グループのプロパティ」ダイアログを開くこともできます。

「スクリプト」。実行時にこのコントロールで生成され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- 値 `%%ThisValue%%` は、ラジオ・ボタン・グループの実行時の値 (選択したラジオ・ボタンの「スクリプト」プロパティの値) を指定します。これはデフォルトです。「スクリプト」プロパティに `%%ThisValue%%` が含まれ、選択したラジオ・ボタンによってスクリプトが生成されない場合、ラジオ・ボタン・グループはスクリプトを生成しません。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

### ラジオ・ボタンの定義

「ラジオ・グループ・ボタンのプロパティ」ダイアログ・ボックスでは、ラジオ・ボタンのグループを指定することができます。

「識別子」。ラジオ・ボタンの固有の識別子です。

「列名」。ラジオ・ボタンの横に表示される名前です。名前は必須フィールドです。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「ニーモニック・キー」。名前に含まれていて、ニーモニックとして使用するオプションの文字です。指定する文字は、名前に含まれていなければなりません。

「ネストされたグループ」。このラジオ・ボタンの下に他のコントロールをネストできるかどうかを指定します。デフォルトは `false` です。「ネストされたグループ」プロパティが `true` に設定されていると、関連するラジオ・ボタンの下に、四角形のドロップ・ゾーンがネストされ、インデントして表示されます。ラジオ・ボタンの下にネストできるコントロールは、フィールド・ピッカー、チェック・ボックス、テキスト・コントロール、静的テキスト、数値コントロール、コンボ・ボックス、リスト・ボックス、およびファイル・ブラウザーです。

「デフォルト」。このラジオ・ボタンがデフォルトの選択かどうかを指定します。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

「スクリプト」。ラジオ・ボタンが選択されたときに作成される R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- ネストされたコントロールを含むラジオ・ボタンについては、`%%ThisValue%%` 値で、ネストされた各コントロールによって生成される R スクリプトまたは Python for Spark スクリプトの空白で区切られたリストが生成されます。リスト内での順序は、ラジオ・ボタンの下に表示される順序になります (上から下)。

既存のリストの下部にある空白行に新しいラジオ・ボタンを追加できます。識別子以外のプロパティのいずれかを入力すると、固有の識別子が生成されます。これは、そのまま使用することもできれば、変更することも可能です。ラジオ・ボタンは、ボタンの「識別子」セルをクリックしてから delete キーを押すことにより削除できます。

## チェック・ボックス・グループ

チェック・ボックス・グループ・コントロールは、1つのチェック・ボックスを使用して、グループとして有効化または無効化されるコントロールのセットのコンテナです。次のタイプのコントロールは、チェック・ボックス・グループに含めることができます: フィールド・ピッカー、チェック・ボックス、コンボ・ボックス、リスト・ボックス、テキスト・コントロール、数値コントロール、静的テキスト、ラジオ・グループ、およびフィールド・ブラウザー。チェック・ボックス・グループ・コントロールには、次のプロパティがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

「**タイトル**」。グループのオプションのタイトル。複数行にわたるタイトルの場合、`¥n` を使用して改行を指定します。

「**チェック・ボックスのタイトル**」。制御するチェック・ボックスと共に表示されるオプション・ラベル。改行を指定する `¥n` をサポートしています。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「**デフォルト値**」。制御するチェック・ボックスのデフォルトの状態 (チェックされた状態、またはチェックされていない状態)。

「**チェックされた/チェックされていない R スクリプト**」。コントロールにチェック・マークが付けられたとき、およびチェック・マークが外されたときに生成される R スクリプト・シンタックスを指定します。スクリプト・テンプレートに R スクリプトを含めるには、「識別子」プロパティの値を使用します。生成された R スクリプトは、「チェックされた R スクリプト」プロパティと、「チェックされていない R スクリプト」プロパティのどちらから生成されたかにかかわらず、識別子の指定した位置に挿入されます。例えば、識別子が `checkboxgroup1` である場合、ランタイム時にスクリプト・テンプレートの `%checkboxgroup1%` のインスタンスが、ボックスにチェック・マークが付けられている場合は「チェックされた R スクリプト」プロパティの値に、ボックスにチェック・マークが付けられていない場合は「チェックされていない R スクリプト」プロパティの値に置き換えられます。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティ・ダイアログにスクリプトを入力します。
- チェック・ボックス・グループに含まれている任意のコントロールの識別子を含めることができます。実行時に、識別子はコントロールで生成された R スクリプトに置き換えられます。
- 値 `%%ThisValue%%` は、「チェックされた R スクリプト」プロパティまたは「チェックされていない R スクリプト」プロパティのいずれかに使用できます。チェック・ボックス・グループの各コントロールで生成された R スクリプトのリストを生成します。このリストは空白で区切られ、上から下へ、グループ内に表示される順に並べられています。
- デフォルトでは、「チェックされた R スクリプト」プロパティの値は `%%ThisValue%%` で、「チェックされていない R スクリプト」プロパティは空白です。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されません。

## ファイル・ブラウザー

ファイル・ブラウザー・コントロールは、ファイル・パスのテキスト・ボックス、およびファイルを開いたり保存したりするための標準的な IBM SPSS Modeler ダイアログを開く参照ボタンで構成されています。ファイル・ブラウザー・コントロールには、次のプロパティーがあります。

「**識別子**」。コントロールの固有の識別子。これはスクリプト・テンプレートの コントロールを参照する際に使用する識別子です。

「**タイトル**」。コントロールの上に表示されるオプションのタイトル。複数行にわたるタイトルの場合、 $\backslash n$  を使用して改行を指定します。

「**表題位置**」。コントロールに対するタイトルの相対位置を指定します。値は「上」または「左」ですが、「上」がデフォルトです。

「**ツールチップ**」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「**ニーモニック・キー**」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「**ファイル・システム操作**」。参照ボタンで起動したダイアログが、ファイルを開くためのダイアログ、またはファイルを保存するためのダイアログのどちらに該当するかを指定します。「開く」の値は、指定したファイルが存在するかどうかを参照ダイアログが検証することを示します。「保存」の値は、指定したファイルが存在するかどうかを参照ダイアログが検証しないことを示します。

「**ブラウザーの種類**」。参照ダイアログを使用してファイルを選択する（「ファイルを選択」）か、またはフォルダーを選択する（「フォルダーを選択」）かを指定します。

「**ファイルの絞り込み条件**」。省略符号 (...) ボタンをクリックして「ファイルの絞り込み条件」ダイアログ・ボックスを開き、「開く」ダイアログまたは「保存」ダイアログで使用可能なファイル・タイプを指定できます。デフォルトでは、すべての種類のファイルが使用できます。キャンバス上のファイル・ブラウザー・コントロールをダブルクリックして、「ファイルの絞り込み条件」ダイアログを開くこともできます。

「**ファイル・システムの種類**」。ディストリビュート・アナリシス・モードで、IBM SPSS Modeler Server が実行されているファイル・システムまたはローカル・コンピューターのファイル・システムのどちらかを、「開く」または「保存」ダイアログが参照するかを指定します。「**サーバー**」を選択してサーバーのファイル・システムを参照するか、または「**クライアント**」を選択してローカル・コンピューターのファイル・システムを参照します。ローカル・アナリシス・モードでは、このプロパティーは無効となります。

「**実行に必要**」。実行を続行するために、このコントロールで値が必要かどうかを指定します。「真」が指定された場合、ノード・ダイアログのユーザーはコントロールに対して値を指定する必要があります。これを行わずに「OK」ボタンをクリックすると、エラーが生成されます。「**False**」が指定された場合、このコントロールの値は「OK」ボタンの状態に影響を与えません。デフォルトは **False** です。

「**デフォルト**」。コントロールのデフォルト値。

「**スクリプト**」。実行時にこのコントロールで生成され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Spark 用 Python スクリプトを指定できます。複数行のスクリプトまたは長いスクリプトの場合、省略符号 (...) ボタンをクリックして、スクリプトのプロパティー・ダイアログにスクリプトを入力します。
- 値 `%%ThisValue%%` は、テキスト・ボックスの実行時の値（手動で指定されるか、または参照ダイアログによって設定される二重引用符で囲まれたファイル・パス）を指定します。これはデフォルトです。
- 「スクリプト」プロパティーに `%%ThisValue%%` が含まれ、テキスト・ボックスのランタイム時の値が空の場合、ファイル・ブラウザー・コントロールはスクリプトを生成しません。

「**有効化ルール**」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティーは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されません。

## ファイル・タイプ・フィルター

「ファイル・フィルター」ダイアログ・ボックスでは、「ファイル・システム・ブラウザ」コントロールからアクセスできるファイルを開くダイアログや保存するダイアログでの「ファイルの種類」および「次の種類で保存」ドロップダウン・リストに表示されるファイルの種類を指定できます。デフォルトでは、すべての種類のファイルが使用できます。

ダイアログ・ボックスに明示的には表示されないファイルの種類を指定するには

1. 「その他」を選択します。
2. ファイルの種類の名前を入力します。
3. \*.suffix という形式を使用してファイルの種類を入力します (例えば、\*.xls)。それぞれをセミコロンで区切って、複数のファイルの種類を指定することもできます。

## タブ

タブ・コントロールは、ノード・ダイアログにタブを追加します。新規タブには、他の任意のコントロールを追加することができます。タブ・コントロールには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。

「タイトル」。タブのタイトル。

**位置:** ノード・ダイアログ内のタブの位置を、ノード・ダイアログ内の他のタブを基準にして指定します。

「スクリプト」。実行時にこのコントロールによって生成および実行され、スクリプト・テンプレートに挿入できる R スクリプトまたは Python for Spark スクリプトを指定します。

- 有効な R スクリプトまたは Python for Spark スクリプトを指定し、改行に  $\backslash n$  を使用できます。
- 値 `%%ThisValue%%` は、タブ内の各コントロールで生成された R スクリプトまたは Python for Spark スクリプトのリストを生成します。このリストは空白で区切られ、タブ内で表示される順 (上から下および左から右) に並べられています。これがデフォルトです。
- 「スクリプト」プロパティに `%%ThisValue%%` が含まれ、タブのどのコントロールによっても R スクリプトも Python for Spark スクリプトも生成されない場合、タブは全体として何のスクリプトも生成しません。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

## サブダイアログ・ボタン

サブダイアログ・ボタン・コントロールは、サブダイアログを起動するボタンを指定し、サブダイアログのダイアログ・ビルダーへのアクセスを提供します。サブダイアログ・ボタンには、次のプロパティがあります。

「識別子」。コントロールの固有の識別子。

「タイトル」。: ボタンに表示されるテキスト。

「ツールチップ」。ユーザーがコントロールにカーソルを移動した場合に表示されるオプションのツールチップ・テキスト。

「サブダイアログ」。省略符号 (...) ボタンをクリックして、サブダイアログのカスタム・ダイアログ・ビルダーを開きます。「サブダイアログ」ボタンをダブルクリックして、ビルダーを開くこともできます。

「ニーモニック・キー」。コントロールへのキーボード・ショートカットとして使用する、タイトルのオプションの文字。これらの文字は、タイトル内で下線付きで表示されます。ショートカットは、Alt キーを押しながらニーモニック・キーを押すとアクティブ化されます。

「有効化ルール」。現在のコントロールが有効になる条件を決定するルールを指定します。省略符号 (...) ボタンをクリックし、ダイアログ ボックスを開き、ルールを指定します。「有効化ルール」プロパティは、有効化ルールを指定するために使用できる他のコントロールがキャンバスに存在するときのみ表示されます。

注: サブダイアログ・ボタン・コントロールをサブダイアログに追加することはできません。

## サブダイアログのダイアログ・プロパティ

サブダイアログのプロパティを表示および設定するには、次のように操作します。

1. メイン・ダイアログのサブダイアログのボタンをダブルクリックしてサブダイアログを開くか、またはサブダイアログ・ボタンをシングルクリックし、サブダイアログのプロパティの省略符号 (...) ボタンをクリックします。
2. サブダイアログで、コントロール外部の領域にあるキャンバスをクリックします。キャンバスにコントロールがない場合、サブダイアログのプロパティは常に表示されます。

「サブダイアログ名」。サブダイアログの固有の識別子。「サブダイアログ名」プロパティは必須です。

注: サブダイアログ名を (%%My Sub-dialog Name%% のように) スクリプト・テンプレートの識別子として指定した場合、ランタイム時に、サブダイアログの各コントロールで生成されたスクリプトのリストに置き換えられます。このリストは空白で区切られ、上から下へ、左から右へと、表示される順に並べられています。

**タイトル**。サブダイアログ・ボックスのタイトル・バーに表示されるテキストを指定します。「タイトル」プロパティはオプションですが、推奨されています。

「ヘルプ・ファイル」。サブダイアログのオプションのヘルプ・ファイルへのパスを指定します。これは、「ヘルプ」ボタンをクリックすると起動するファイルで、メイン・ダイアログに対して指定したヘルプ・ファイルと同じである場合があります。ヘルプ・ファイルは HTML 形式でなければなりません。詳しくは、『ダイアログのプロパティ』の「ヘルプ・ファイル」プロパティの説明を参照してください。

## コントロールの有効化ルールの指定

コントロールが有効になる条件を決定するルールを指定できます。例えば、フィールドピッカーに値が入力されるとラジオ・グループが有効になるように指定できます。有効化ルールの指定に使用できるオプションは、ルールを定義するコントロールのタイプによって異なります。

### フィールド・ピッカー

現在のコントロールが、フィールド・ピッカーに少なくとも 1 つのフィールドが取り込まれると (空でなくなると) 有効になるように指定できます。また現在のコントロールが、フィールド・ピッカーに値が取り込まれないと (空である) 有効になるように指定することもできます。

### チェック・ボックスまたはチェック・ボックス・グループ

現在のコントロールが、チェック・ボックスまたはチェック・ボックス・グループがオンになると有効になるように指定できます。また現在のコントロールが、チェック・ボックスまたはチェック・ボックス・グループがオフになると有効になるように指定することもできます。

### コンボ・ボックスまたは単一選択リスト・ボックス

現在のコントロールが、コンボ・ボックスまたは単一選択リスト・ボックスで特定の値が選択されると有効になるように指定できます。または、コンボ・ボックスまたは単一選択リスト・ボックスで特定の値が選択されていないと、現在のコントロールが有効になるように指定できます。

### 複数選択リスト・ボックス

現在のコントロールが複数選択リスト・ボックスで選択された値の中に特定の値が含まれていると有効になるように指定できます。あるいは、複数選択リスト・ボックスで選択された値の中に特定の値が含まれていないと、現在のコントロールが有効になるように指定できます。

### ラジオ・グループ

現在のコントロールが、特定のラジオ・ボタンが選択されていると有効になるように指定できます。あるいは、特定のラジオ・ボタンが選択されていないと、現在のコントロールが有効になるように指定できます。

有効化ルールを指定できるコントロールには、関連付けられた有効化ルールのプロパティがあります。

注:

- 有効化ルールは、ルールを定義するコントロールが有効になっているかいないかにかかわらず適用されます。例えば、フィールドピッカーに値が設定されるとラジオグループが有効になるように指定するルールがあるとします。このラジオ・グループは、フィールドピッカーが有効であるかどうかに関係なく、フィールドピッカーに値が取り込まれるたびに有効になります。

- タブ・コントロールが無効になっていると、そのタブ上のコントロールは、それらのいずれかに条件が満たされている有効化ルールがあるかどうかに関係なく、すべて無効になります。
- チェック・ボックス・グループが無効になっている場合、グループ内のコントロールは、制御するチェック・ボックスがオンになっているかに関係なく、すべて無効になります。

## 拡張プロパティ

「拡張プロパティ」ダイアログでは、拡張の名前や拡張内のファイルなどの、拡張用のカスタム ダイアログビルダー内の現在の拡張に関する情報を指定します。

- 拡張用のカスタム ダイアログ ビルダーで作成されるすべてのカスタム ノード ダイアログは、ある拡張の一部になります。
- 「拡張プロパティ」ダイアログの「必須」タブにあるフィールドは、拡張とそれに含まれるカスタム ノード ダイアログをインストールする前に指定する必要があります。

拡張のプロパティを指定するには、拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「拡張」 > 「プロパティ」

### 拡張の必須プロパティ

#### 名前

拡張に関連付ける固有の名前。最大 3 つの単語を指定でき、大文字と小文字が区別されません。文字は 7 ビットの ASCII に制限されます。名前の競合の可能性を最小限にするには、複数の単語から成る名前を使用して、最初の単語を URL などの識別子にすることもできます。拡張の保存時に、拡張バンドル (.mpe) のファイル名にもデフォルトでこの名前が使用されることに注意してください。保存時にデフォルト名を使用することをお勧めします。別の名前では保存すると、拡張を今後アンインストールできなくなります。

#### 要約

拡張の簡単な説明で、1 行で表示されます。

#### バージョン

バージョンの識別子。形式は x.x.x で、識別子の各構成要素は整数でなければなりません (例: 1.0.0)。指定されていない場合、0 と見なされます。例えば、3.1 というバージョン識別子は 3.1.0 を意味します。バージョン識別子は、IBM SPSS Modeler バージョンとは無関係です。

#### SPSS Modeler バージョンの最低条件

拡張機能を実行するために必要な最小 SPSS Modeler バージョン。

#### ファイル

「ファイル」リストには、現在拡張に含まれているファイルが表示されます。拡張にファイルを追加するには、「追加」をクリックします。拡張からファイルを削除することも、指定したフォルダにファイルを抽出することもできます。

- カスタム ノード ダイアログのファイル・タイプは .cfe です。
- 拡張のコンポーネントに対する翻訳ファイルは、「省略可能」タブの「ローカライズ」設定で追加します。
- 拡張には、ReadMe ファイルを追加できます。ファイル名には ReadMe.txt を指定します。ユーザーは、拡張の詳細を表示するダイアログから ReadMe ファイルにアクセスできます。ReadMe\_<言語識別子>.txt のように指定することで、readme ファイルのローカライズ版を追加できます。例えば、フランス語バージョンは、ReadMe\_fr.txt となります。

### 拡張のオプションのプロパティ

#### 一般プロパティ

##### 説明

「要約」フィールドに表示するものより詳細な、拡張の説明。例えば、拡張で使用できる主な機能をリストアップすることが考えられます。

## 日付

拡張の現在のバージョンのオプションの日付。形式は指定されません。

## 作成者

拡張の作成者。E メールアドレスを含めることが考えられます。

## リンク

拡張に関連付ける URL のセット。例えば、作成者のホーム・ページなどです。このフィールドの形式は任意です。そのため、スペース、コンマ、またはその他の適切な区切り文字を使用して、複数の URL を区切るようにしてください。

## キーワード

拡張に関連付けるために使用するキーワードのセット。

## プラットフォーム

特定のオペレーティング システム プラットフォームでの拡張の使用に適用される制限に関する情報。

## 従属関係

### SPSS Modeler バージョンの最高条件

拡張を実行できる IBM SPSS Modeler の最大バージョン。

### Integration Plug-in for R が必須

Integration Plug-in for R が必須であるかどうかを指定します。

拡張が CRAN リポジトリのいずれかの R パッケージを必要とする場合、「必須の R パッケージ」コントロールにそれらのパッケージの名前を入力します。名前は大文字小文字が区別されます。最初のパッケージを追加するには、「必須の R パッケージ」コントロール内のどこかをクリックして、入力フィールドを強調表示にします。特定の行にカーソルを置いて **Enter** キーを押すと、新しい行が作成されます。行を選択して **Delete** キーを押すと、その行が削除されます。

## ローカライズ

### カスタム ノード

プロパティ ファイルの翻訳バージョン (ノード ダイアログに表示されるすべての文字列を指定するもの) を、拡張内のカスタム ノード ダイアログに追加できます。特定のノード ダイアログに対する翻訳を追加するには、「[翻訳の追加](#)」をクリックし、翻訳バージョンを格納するフォルダを選択します。特定のノード ダイアログに対するすべての翻訳ファイルを、同じフォルダに配置する必要があります。翻訳ファイルの作成については、[54 ページの『ローカライズされたカスタム・ノード・ダイアログの作成』](#)のトピックを参照してください。

### 翻訳カタログ フォルダ

エンド ユーザーが拡張ハブから拡張の詳細を表示したときに表示される、拡張の「**要約**」フィールドおよび「**説明**」フィールドのローカライズ版を提供できます。拡張のローカライズされたすべてのファイルのセットは、lang という名前のフォルダに配置する必要があります。ローカライズされたファイルが含まれている lang フォルダーを参照し、そのフォルダーを選択します。

ローカライズされたバージョンの「**要約**」および「**説明**」フィールドを提供するには、翻訳を行う言語ごとに、<extension name>\_<language-identifier>.properties という名前のファイルを作成します。実行時に現在のユーザー インターフェイス 言語の .properties ファイルが見つからない場合は、「**必須**」タブおよび「**省略可能**」タブで指定した「**要約**」フィールドおよび「**説明**」フィールドの値が使用されます。

- <extension name> は、拡張機能の「**名前**」フィールドの値ですが、スペースはすべてアンダースコアに置き換えます。
- <language-identifier> は、特定の言語の識別子です。IBM SPSS Modeler でサポートされている言語の識別子を後述します。

例えば、MYORG MYSTAT という拡張のフランス語翻訳は、MYORG\_MYSTAT\_fr.properties ファイルに保存されます。

.properties ファイルには次の 2 行が含まれています。これらの行は、2 つのフィールドのローカライズされたテキストを指定しています。

```
Summary=<localized text for Summary field>  
Description=<localized text for Description field>
```

- キーワードである **Summary** と **Description** は英語で記述する必要があり、ローカライズされたテキストはキーワードと同じ行に改行なしで指定する必要があります。
- ファイルは ISO 8859-1 エンコードである必要があります。このエンコードで直接表現できない文字は、Unicode のエスケープ (「¥u」) を使用して記述する必要があります。

ローカライズされたファイルを含む lang フォルダーには、<language-identifier> を名前に持つサブフォルダーが必要です。このサブフォルダーには、特定の言語のローカライズされた .properties ファイルが含まれます。例えば、フランス語の .properties ファイルは lang/fr フォルダーに含まれている必要があります。

言語識別子

**de**。ドイツ語

**en**。英語

**es**。スペイン語

**fr**。フランス語

**it**。イタリア語

**ja**。日本語

**ko**。韓国語

**pl**。ポーランド語

**pt\_BR**。ブラジル・ポルトガル語

**ru**。ロシア語

**zh\_CN**。中国語 (簡体字)

**zh\_TW**。中国語 (繁体字)

## カスタム・ノード・ダイアログの管理

拡張用のカスタム ダイアログ ビルダーを使用すると、拡張内で自分や他のユーザーが作成したカスタム ノード ダイアログを管理できます。カスタム・ノード・ダイアログを使用するには、それらを必要とする SPSS Modeler Client または SPSS Modeler Batch のすべてのインスタンスにインストールしなければなりません。カスタム・ダイアログ・ノードをサーバー・モードで使用する場合、SPSS Modeler Server に何もインストールする必要はありません。

注: 変更できるカスタム・ノード・ダイアログは、IBM SPSS Modeler で作成されたもののみです。

### カスタム ノード ダイアログが含まれる拡張を開く

1 つ以上のカスタム ノード ダイアログの仕様が含まれる拡張バンドルファイル (.mpe) を開くことも、インストールされた拡張を開くこともできます。拡張内の任意のノード ダイアログを変更し、拡張を保存またはインストールできます。拡張をインストールすると、その拡張に含まれているノード ダイアログがインストールされます。拡張を保存すると、その拡張内の任意のノード ダイアログに加えられた変更内容が保存されます。

拡張バンドルファイルを開くには、拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「開く」

インストールされた拡張を開くには、拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。



「ファイル」 > 「インストールされた項目を開く」

注：インストールされた拡張を変更するために開いている場合は、「ファイル」 > 「インストール」を選択すると再インストールされ、既存のバージョンが置き換えられます。カスタム・ダイアログ・ビルダーを使用して作成したノードのコンテキスト・メニューで「編集」を使用しても、カスタム・ダイアログ・ビルダーのノード・ダイアログは開きません。

### 拡張バンドルファイルへの保存

拡張用のカスタム ダイアログ ビルダーで開いている拡張を保存すると、その拡張に含まれているカスタム ノード ダイアログも保存されます。拡張は、拡張バンドルファイル (.mpe) に保存されます。「拡張プロパティ」ダイアログの「名前」フィールドで指定した名前と一致するデフォルトのファイル名を保持することをお勧めします。

拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「保存」

### 拡張のインストール

拡張用のカスタム ダイアログ ビルダーで開いている拡張をインストールすると、その拡張に含まれているカスタム ノード ダイアログもインストールされます。既存の拡張をインストールすると、既存のバージョンが置き換えられます。これには、既にインストールされていた拡張内のすべてのカスタム ノード ダイアログも含まれます。

現在開いている拡張をインストールするには、拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「インストール」

デフォルトでは、拡張は、ご使用のオペレーティング システムの、一般ユーザーが書き込み可能な場所にインストールされます。詳しくは、[26 ページの『拡張のインストール場所』](#)のトピックを参照してください。

注：開いているストリームでは、拡張内に含まれているノード ダイアログの既存のバージョンは置き換えられません。再インストールされたカスタム・ダイアログ・ビルダー・ノードが含まれるストリームを開くと、警告メッセージを受け取ります。

### 拡張のアンインストール

拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「アンインストール」

拡張をアンインストールすると、その拡張に含まれているすべてのカスタム ノード ダイアログがアンインストールされます。拡張ハブでアンインストールすることもできます。

注：拡張を正常にアンインストールするには、拡張バンドル .mpe のファイル名が「拡張プロパティ」ダイアログで指定された名前と一致している必要があります。これはデフォルトのファイル名です。ファイル名を変更した場合は、「名前」フィールドに一致する名前に変更し、再度アンインストールを試みてください。

### カスタム ダイアログ パッケージ ファイルのインポート

拡張用のカスタム ダイアログ ビルダーに、カスタム ダイアログ パッケージ (.cfd) ファイルをインポートできます。.cfd ファイルは、.cfe ファイルに変換され、これが新しい拡張に追加されます。

拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「インポート」

拡張用のカスタム ダイアログ ビルダー内で、「拡張」 > 「プロパティ」を使用してアクセスする「拡張プロパティ」ダイアログから、拡張に .cfe ファイルを追加することもできます。

## 拡張へのカスタム ノード ダイアログの追加

拡張に、新しいカスタム ノード ダイアログを追加できます。

拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「拡張」 > 「新規ダイアログ」

## 拡張内の複数のカスタム ノード ダイアログ間の切り替え

現在の拡張に、複数のカスタム ノード ダイアログが含まれている場合、それらを切り替えることができます。

拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「拡張」 > 「ダイアログの編集」。次に、処理するカスタム ノード ダイアログを選択します。

## 新しい拡張の作成

拡張用のカスタム ダイアログ ビルダーで新しい拡張を作成すると、その拡張に、新しい空のカスタム ノード ダイアログが追加されます。

新しい拡張を作成するには、拡張用のカスタム ダイアログ ビルダーのメニューで、次のように選択します。

「ファイル」 > 「新規作成」

## SPSS Modeler Batch または IBM SPSS Collaboration and Deployment Services 内の拡張

SPSS Modeler Batch または IBM SPSS Collaboration and Deployment Services のインストール済み環境内の拡張を使用するには、環境変数 `IBM_SPSS_MODELER_EXTENSION_PATH` が対象の環境に定義されていて、それが拡張を格納している場所を指していることを確認してください。

`IBM_SPSS_MODELER_EXTENSION_PATH` 環境変数を定義する前に、カスタム・ノードを含むストリームが IBM SPSS Collaboration and Deployment Services Repository に格納された場合、このストリームを正常に実行するには、リポジトリに再格納する必要があります。

注：SPSS Modeler Batch または IBM SPSS Collaboration and Deployment Services Adapter for SPSS Modeler のバージョンが、拡張機能が作成された SPSS Modeler Client のバージョンと一致していることを確認してください。

## ローカライズされたカスタム・ノード・ダイアログの作成

IBM SPSS Modeler でサポートされている任意の言語に対して、ローカライズされたカスタム・ノード・ダイアログを作成できます。カスタム・ノード・ダイアログに表示される 任意の文字列をローカライズでき、また、オプションのヘルプ・ファイルをローカライズできます。

### ダイアログの文字列をローカライズするには

デプロイを予定している言語ごとに、カスタム ノード ダイアログに関連付けられているプロパティ ファイルを作成し、そのコピーを作成する必要があります。プロパティ・ファイルには、ノード・ダイアログに関連付けられたローカライズ可能なすべての文字列が含まれています。

「拡張プロパティ」ダイアログ (拡張用のカスタム ダイアログ ビルダー内) で、ファイルを選択し、「抽出の基準」をクリックすることで、拡張からカスタム ノード ダイアログ ファイル (.cfe) を抽出します。次に、.cfe ファイルの内容を抽出します。.cfe ファイルは、単純な .zip ファイルです。.cfe ファイルから抽出された内容には、サポート対象言語ごとのプロパティ ファイルが含まれています。ここで、特定の言語のファイルの名前は、`<Dialog Name>_<language identifier>.properties` になります (言語識別子は、次の表を参照)。

1. 翻訳する予定の各プロパティ ファイルを、UTF-8 をサポートするテキスト エディタ (Windows の Notepad など) で開きます。ローカライズする必要があるプロパティに関連付けられた値を変更します。ただし、プロパティの名前は変更しません。特定のコントロールに関連付けられたプロパティには、コントロールの識別子の接頭辞が付加されます。例えば、識別子 `options_button` を持つコント

ールの「ツールチップ」プロパティは、*options\_button\_tooltip\_LABEL* となります。「タイトル」プロパティの名前は、単に *<identifier>\_LABEL* (例: *options\_button\_LABEL*) となります。

2. 「拡張プロパティ」ダイアログの「省略可能」タブの「ローカライズ」設定で、プロパティ ファイルのローカライズ版をカスタム ノード ダイアログ ファイル (.cfe) に追加します。詳しくは、[50 ページの『拡張のオプションのプロパティ』](#)のトピックを参照してください。

ノード ダイアログの起動時に、IBM SPSS Modeler は、「オプション」ダイアログ ボックスの「全般」タブにある「言語」ドロップダウンで指定したとおり、言語の識別子が現在の言語と一致するプロパティ ファイルを検索します。該当するプロパティ ファイルが見つからない場合は、デフォルト ファイル *<Dialog Name>.properties* が使用されます。

## ヘルプ・ファイルをローカライズするには

1. カスタム ノード ダイアログに関連付けられているヘルプ ファイルのコピーを作成し、テキストを目的の言語にローカライズします。
2. 次の表にある言語の識別子を使用して、コピーを *<Help File>\_<language identifier>.htm* に名前変更します。例えば、ヘルプ・ファイルが *myhelp.htm* で、ドイツ語バージョンのファイルを作成する場合、ローカライズされたヘルプ・ファイルの名前は *myhelp\_de.htm* にする必要があります。

非ローカライズ版と同じディレクトリに、ヘルプ ファイルのすべてのローカライズ版を格納します。「ダイアログ プロパティ」の「ヘルプ ファイル」プロパティから非ローカライズ版のヘルプ ファイルを追加すると、ローカライズ版が自動的にノード ダイアログに追加されます。

ローカライズする必要がある補足ファイル (イメージ ファイルなど) がある場合、メイン ヘルプ ファイルの該当するパスを、ローカライズ版を指すように手動で変更する必要があります。ローカライズ版を含む補足ファイルは、手動でカスタム ノード ダイアログ (.cfe) ファイルに追加する必要があります。カスタム ノード ダイアログ ファイルにアクセスして手動で変更する方法については、『ダイアログの文字列をローカライズするには』というタイトルの前セクションを参照してください。

ノード ダイアログの起動時に、IBM SPSS Modeler は、ノード 「オプション」ダイアログ ボックスの「全般」タブにある「言語」ドロップダウンで指定したとおり、言語の識別子が現在の言語と一致するヘルプ ファイルを検索します。該当するヘルプ・ファイルがない場合は、ノード・ダイアログに対して指定したヘルプ・ファイル (「ダイアログ・プロパティ」の「ヘルプ・ファイル」プロパティで指定) が使用されます。

言語識別子

**de.** ドイツ語

**en.** 英語

**es.** スペイン語

**fr.** フランス語

**it.** イタリア語

**ja.** 日本語

**ko.** 韓国語

**pl.** ポーランド語

**pt\_BR.** ブラジル・ポルトガル語

**ru.** ロシア語

**zh\_CN.** 中国語 (簡体字)

**zh\_TW.** 中国語 (繁体字)

注: カスタム・ノード・ダイアログおよび関連するヘルプ・ファイルのテキストは、IBM SPSS Modeler でサポートされている言語に限定されません。言語固有のプロパティおよびヘルプ・ファイルを作成することなく、ノード・ダイアログおよびヘルプ・テキストを任意の言語で自由に記述できます。これにより、ノード・ダイアログのすべてのユーザーは、その言語でテキストを見ることができます。

## Python for Spark を使用したデータのインポートおよびエクスポート

拡張用のカスタム ダイアログ ビルダーを使用して、カスタム ノードを作成し、Python for Spark スクリプトを作成して、データ ソースがある任意の場所からデータを読み取り、Apache Spark によってサポートされている任意のデータ形式にデータを書き込むことができます。

例えば、ユーザーがデータをデータベースに書き込む場合を考えてみます。そのユーザーは、拡張用のカスタム ダイアログ ビルダーおよび Python for Spark を使用してカスタム エクスポート JDBC ノードを作成し、そのモデルを実行してデータをデータベースに書き込みます。データをデータベースから読み取るために、カスタム インポート JDBC ノードを作成することもできます。また、この同じ方法を使用して、例えば JSON ファイルからデータを SPSS Modeler に読み込ませることができます。これにより、このユーザーはデータを SPSS Modeler に読み込ませた後で、使用可能なすべての SPSS Modeler ノードを使用してビジネス上の問題に取り組むことができます。

注：Python for Spark のインポートおよびエクスポートの機能とともに JDBC を使用する場合は、ご使用の JDBC ドライバ ファイルを IBM SPSS Modeler インストール ディレクトリ内の as/lib ディレクトリにコピーする必要があります。

### Python for Spark を使用したデータのインポート/エクスポート

1. 「拡張機能」 > 「カスタム ノード ダイアログ ビルダー」に移動します。
2. 「ダイアログ プロパティ」の下で、スクリプトタイプとして「Python for Spark」を、ノードタイプとして「インポート」または「エクスポート」を選択します。
3. 必要に応じて、ダイアログ名などのその他のプロパティを入力します。
4. 「スクリプト」セクションで、データのインポートまたはエクスポートのための Python for Spark スクリプトを入力するか、貼り付けます。
5. 「インストール」をクリックして、Python for Spark スクリプトをインストールします。新しいカスタム・インポート・ノードが「入力」パレットに追加され、新しいカスタム・エクスポート・ノードが「エクスポート」パレットに追加されます。

## R を使用したデータのインポートおよびエクスポート

拡張用のカスタム ダイアログ ビルダーを使用して、カスタム ノードを作成し、R スクリプトを作成して、データ ソースがある任意の場所からデータを読み取り、R によってサポートされている任意のデータ形式にデータを書き込むことができます。

例えば、ユーザーがデータをデータベースに書き込む場合を考えてみます。そのユーザーは、拡張用のカスタム ダイアログ ビルダーおよび R スクリプトを使用してカスタム エクスポート JDBC ノードを作成し、そのモデルを実行してデータをデータベースに書き込みます。データをデータベースから読み取るために、カスタム インポート JDBC ノードを作成することもできます。この同じ方法を使用して、JSON ファイルなどからデータを SPSS Modeler に読み込ませることもできます。これにより、このユーザーはデータを SPSS Modeler に読み込ませた後で、使用可能なすべての SPSS Modeler ノードを使用してビジネス上の問題に取り組むことができます。

### R を使用したデータのインポート/エクスポート

1. 「拡張機能」 > 「カスタム ノード ダイアログ ビルダー」に移動します。
2. 「ダイアログ プロパティ」の下で、スクリプトタイプとして「R」を、ノードタイプとして「インポート」または「エクスポート」を選択します。
3. 必要に応じて、ダイアログ名などのその他のプロパティを入力します。
4. 「スクリプト」セクションで、データのインポートまたはエクスポートのための R スクリプトを入力するか、貼り付けます。
5. 「インストール」をクリックして、R スクリプトをインストールします。新しいカスタム・インポート・ノードが「入力」パレットに追加され、新しいカスタム・エクスポート・ノードが「エクスポート」パレットに追加されます。

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 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

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

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。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、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。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしく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保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive, MD-NC119*

*Armonk, NY 10504-1785*

*US*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。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者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、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。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

## 商標

---

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

## 製品資料に関するご使用条件

---

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

### 適用範囲

IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

### 個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布（頒布、送信を含む）または表示（上映を含む）することはできません。

### 商業的使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。



## 権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入 関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。





# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。  
なお, 濁音と半濁音は清音と同等に扱われています。

## [カ行]

### 拡張機能

新しい拡張機能の検出とインストール [24](#)

インストール済みの拡張機能の表示 [25](#)

拡張機能の更新のインストール [24](#)

拡張機能の削除 [25](#)

拡張機能の詳細 [25](#)

### 拡張のインポート・ノード

「コンソール出力」タブ [20](#)

### 拡張のエクスポート・ノード

「コンソール出力」タブ [12](#)

### 拡張ノード [12](#)

### 拡張の出力ノード

「コンソール出力」タブ [15](#)

出力タブ [15](#)

「シンタックス」タブ [14](#)

### 拡張の出力ブラウザ [15, 16](#)

### 拡張の変換ノード

「コンソール出力」タブ [20](#)

### 拡張バンドル

拡張バンドルのインストール [26](#)

### 拡張モデル・ナゲット

概要 [17](#)

「グラフ出力」タブ [18](#)

「コンソール出力」タブ [19](#)

「シンタックス」タブ [17](#)

「テキスト出力」タブ [18](#)

「モデル・オプション」タブ [17](#)

### 拡張モデル・ノード

「コンソール出力」タブ [16](#)

「テキスト出力」タブ [16](#)

「モデル オプション」タブ [16](#)

### 拡張用のカスタム・ダイアログ・ビルダー

インストールされた拡張のノード ダイアログの変更 [52](#)

拡張バンドルファイル [52](#)

ノード ダイアログが含まれる拡張のインストール [52](#)

ノード ダイアログが含まれる拡張の保存 [52](#)

ノード ダイアログが含まれる拡張を開く [52](#)

### カスタム・ダイアログ・ビルダー

カラー・ピッカー [41](#)

項目グループ・コントロール [44](#)

コンボ・ボックス [34](#)

コンボ・ボックスのリスト項目 [35](#)

サブダイアログのプロパティ [49](#)

サブダイアログ・ボタン [48](#)

数値コントロール [38](#)

スクリプト・テンプレート [30](#)

静的テキスト・コントロール [41](#)

ダイアログおよびヘルプ・ファイルのローカライズ [54](#)

ダイアログのプロパティ [29](#)

タブ [48](#)

チェック・ボックス [34](#)

チェック・ボックス・グループ [46](#)

テーブル コントロール [42](#)

### カスタム・ダイアログ・ビルダー (続き)

テーブル・コントロールの列 [43](#)

テキスト・コントロール [37](#)

日付コントロール [39](#)

ファイル・タイプ・フィルター [48](#)

ファイル・ブラウザ [47](#)

フィールド・ソース [33](#)

フィールド・ピッカー [32](#)

フィールド・リストのフィルタリング [34](#)

ヘルプ・ファイル [29](#)

保護テキスト [40](#)

有効化ルール [49](#)

ラジオ・グループ [44](#)

ラジオ・グループ・ボタン [45](#)

リスト・ボックス [36](#)

リスト・ボックスのリスト項目 [35](#)

レイアウト・ルール [29](#)

preview [31](#)

Python for Spark データ インポート [56](#)

Python for Spark データ エクスポート [56](#)

R データのインポート [56](#)

R データのエクスポート [56](#)

### 「グラフ出力」タブ

拡張モデル・ナゲット [18](#)

### 「コンソール出力」タブ

拡張モデル・ナゲット [19](#)

## [サ行]

### 「シンタックス」タブ

拡張モデル・ナゲット [17](#)

### ストリームの公開

IBM SPSS Modeler Solution Publisher [13](#)

## [タ行]

### 「テキスト出力」タブ

拡張モデル・ナゲット [18](#)

## [ラ行]

利用可能な R シンタックス [1](#)

## A

Apache Spark [1](#)

## I

IBM SPSS Modeler Solution Publisher [13](#)

IBM SPSS Modeler 拡張ノード [12](#)

IBM SPSS Modeler の拡張機能 [1](#)

## P

Python for Spark

Python for Spark (続き)  
API [2](#), [4](#), [7-9](#)

## R

R  
許容される構文 [1](#)

### [特殊文字]

.par ファイル [13](#)  
.pim ファイル [13](#)



