

*IBM SPSS Modeler 18.2.2 Solution  
Publisher*



## 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、[35 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ていない限り、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 18.2.2 Solution Publisher

### 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

|   |           |
|---|-----------|
| 前書き.....  | v         |
| <b>第 1 章 IBM SPSS Modeler について .....</b>  | <b>1</b>  |
| IBM SPSS Modeler 製品.....  | 1         |
| IBM SPSS Modeler .....  | 1         |
| IBM SPSS Modeler Server .....   | 1         |
| IBM SPSS Modeler Administration Console .....   | 2         |
| IBM SPSS Modeler Batch .....  | 2         |
| IBM SPSS Modeler Solution Publisher .....   | 2         |
| IBM SPSS Collaboration and Deployment Services 用の IBM SPSS Modeler Server アダプター ..... | 2         |
| IBM SPSS Modeler のエディション.....   | 2         |
| 資料.....   | 3         |
| SPSS Modeler Professional ドキュメント.....   | 3         |
| SPSS Modeler Premium ドキュメント.....  | 4         |
| アプリケーションの例.....   | 4         |
| Demos フォルダー.....  | 4         |
| ライセンスの追跡.....   | 4         |
| <b>第 2 章 IBM SPSS Modeler Solution Publisher .....</b>                                | <b>5</b>  |
| IBM SPSS Modeler Solution Publisher の動作方法.....  | 5         |
| ストリームの公開.....   | 6         |
| <b>第 3 章 公開済みストリームの実行.....</b>  | <b>9</b>  |
| パラメーター・ファイルを使用したストリーム実行のカスタマイズ.....   | 10        |
| IBM SPSS Modeler Solution Publisher のアプリケーションへの埋め込み.....                              | 12        |
| IBM SPSS Modeler Solution Publisher サンプル・アプリケーション.....                                | 13        |
| <b>第 4 章 IBM SPSS Modeler Solution Publisher Runtime のインストール .....</b>                | <b>15</b> |
| Runtime のインストール.....  | 15        |
| Windows へのインストール.....   | 15        |
| UNIX へのインストール.....  | 15        |
| インストールのトラブルシューティング .....  | 17        |
| Runtime の起動.....  | 17        |
| 一時ディレクトリーの変更.....   | 18        |
| IBM SPSS Modeler Solution Publisher Runtime のアンインストール.....                            | 18        |
| Windows のアンインストール手順.....  | 18        |
| UNIX のアンインストール手順.....   | 18        |
| <b>第 5 章 IBM SPSS Embedded Predictive Modeling API .....</b>                          | <b>19</b> |
| <b>第 6 章 IBM SPSS Modeler Solution Publisher Runtime Library API リファレンス....</b>       | <b>21</b> |
| はじめに.....   | 21        |
| API プロセスの概要.....  | 21        |
| API 関数.....   | 22        |
| <b>特記事項.....</b>  | <b>35</b> |
| 商標.....   | 36        |
| 製品資料に関するご使用条件.....  | 36        |

索引.....39

## 前書き

---

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

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

### IBM Business Analytics について

IBM Business Analytics ソフトウェアは、意思決定者がビジネス・パフォーマンスを向上させるために信頼する完全で、一貫した正確な情報を提供します。ビジネス・インテリジェンス、予測分析、財務実績および戦略管理、分析アプリケーション の包括的なポートフォリオを利用することによって、現在の実績を明確、迅速に理解し、将来の結果を予測することができます。豊富な産業用ソリューション、証明された実践法、それに専門家によるサービスを組み合わせることにより、あらゆる規模の会社組織が、最高の生産性を推進し、信頼できる意思決定を自動化し、そして、よりよい結果を実現させることができます。

このポートフォリオの一部として、IBM SPSS Predictive Analytics ソフトウェアを使用する組織は、将来のイベントを予測し、その洞察に基づいて積極的に行動し、より優れた業績を実現することができます。全世界の企業、政府、学術分野のお客様が IBM SPSS の技術を活用し、不正行為を減少させ、リスクを軽減させながら、顧客の獲得、保持、成長において、競争優位を高めることができます。組織は、IBM SPSS ソフトウェアを日常業務に組み込むことにより、予測力を持つ企業になり、意思決定の管理と自動化を可能にすることで、ビジネス目標を達成し、重要な競争上の優位性を実現します。詳細な情報、または営業担当者へのお問い合わせ方法については、<https://www.ibm.com/mysupport/s/> を参照してください。

### 技術サポート

お客様はテクニカル・サポートをご利用いただけます。IBM Corp. 製品の使用方法、または対応するハードウェア環境へのインストールについてサポートが必要な場合は、テクニカル・サポートにご連絡ください。テクニカル・サポートのご利用には、<https://www.ibm.com/mysupport/s/> の IBM Corp. Web サイトをご覧ください。支援を要請される場合は、事前にユーザー、会社組織、そして、サポート契約を明確にしておいていただくよう、お願いします。



# 第 1 章 IBM SPSS Modeler について

IBM SPSS Modeler は、ビジネスの専門知識を活用して予測モデルを迅速に作成したり、また作成したモデルをビジネス・オペレーションに展開して意思決定を改善できるようにする、一連のデータ・マイニング・ツールです。IBM SPSS Modeler は業界標準の CRISP-DM モデルをベースに設計されたものであり、データ・マイニング・プロセス全体をサポートして、データに基づいてより良いビジネスの成果を達成できるようにします。

IBM SPSS Modeler には、マシン学習、人工知能、および統計に基づいたさまざまなモデル作成方法が用意されています。「モデル作成」パレットを利用して、データから新しい情報を引き出したり、予測モデルを作成することができます。各手法によって、利点や適した問題の種類が異なります。

SPSS Modeler は、スタンドアロン製品として購入または SPSS Modeler Server と組み合わせてクライアントとして使用することができます。後のセクションで説明されているとおり、多くの追加オプションも使用することができます。詳しくは、<https://www.ibm.com/analytics/us/en/technology/spss/> を参照してください。

## IBM SPSS Modeler 製品

製品と関連するソフトウェアの IBM SPSS Modeler ファミリーの構成は次のとおりです。

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (IBM SPSS Deployment Manager に付属)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Collaboration and Deployment Services 用の IBM SPSS Modeler Server アダプター

## IBM SPSS Modeler

SPSS Modeler はこの製品のすべての機能を搭載したバージョンであり、ユーザーのパーソナル・コンピューターにインストールし、そのコンピューターで実行します。スタンドアロン製品としてローカル・モードで SPSS Modeler を実行するか、大規模なデータ・セットを使用する場合にパフォーマンスを向上させるために IBM SPSS Modeler Server と組み合わせて実行することができます。

SPSS Modeler を使用して、プログラミングの必要なく、正確な予測モデルを迅速かつ直感的に構築することができます。独自のビジュアル・インターフェースを使用すると、データ・マイニング・プロセスを簡単に視覚化することができます。製品に組み込まれている高度な分析の支援を受けて、データ内に隠れたパターンやトレンドを発見することができます。結果をモデル化し、ビジネスチャンスを活用してリスクを軽減できるようになり、それらに影響を与える要因を理解することができます。

SPSS Modeler は SPSS Modeler Professional および SPSS Modeler Premium の 2 つのエディションで使用できます。詳しくは、[2 ページの『IBM SPSS Modeler のエディション』](#) のトピックを参照してください。

## IBM SPSS Modeler Server

SPSS Modeler は、クライアント/サーバー・アーキテクチャーを使用して、リソース集中型の操作が必要な要求を、強力なサーバー・ソフトウェアへ分散します。

SPSS Modeler Server は、1 つまたは複数の IBM SPSS Modeler のインストールと組み合わせてサーバー・ホストで分散分析モードで継続的に実行する、別途ライセンスが必要な製品です。このように、SPSS Modeler Server では、メモリー集中型の操作を、クライアント コンピューターにデータをダウンロードせずにサーバー上で実行できるため、大きなデータ・セットで優れたパフォーマンスを発揮します。IBM

SPSS Modeler Server は、パフォーマンスと自動化のさらなる利点を提供し、SQL の最適化とデータベース内のモデリング機能をサポートしています。

## IBM SPSS Modeler Administration Console

Modeler Administration Console は、SPSS Modeler Server 構成オプションの多くを管理するグラフィカル・ユーザー・インターフェースです。それらの構成オプションは、オプション・ファイルで設定することも可能です。コンソールは、IBM SPSS Deployment Manager に含まれています。コンソールを使用すると、SPSS Modeler Server インストール済み環境をモニターしたり、構成したりできます。SPSS Modeler Server の現在の顧客は、コンソールを無料で利用できます。アプリケーションは Windows コンピューターにのみインストールできますが、サポートされる任意のプラットフォームにインストールされたサーバーを管理できます。

## IBM SPSS Modeler Batch

データマイニングは、通常、対話型のプロセスですが、グラフィカル・ユーザー・インターフェースを必要とせずに、コマンドラインから SPSS Modeler を実行することも可能です。例えば、ユーザーの介入なしで実行する長期実行または反復的なタスクがあります。SPSS Modeler Batch は、通常ユーザー・インターフェースにアクセスせずに SPSS Modeler の完全な分析機能のサポートを提供する製品の特別バージョンです。SPSS Modeler Batch を使用するには、SPSS Modeler Server が必要です。

## IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher は、外部ランタイムで実行することができ、外部アプリケーションに埋め込まれる SPSS Modeler ストリームのパッケージ版を作成することができるツールです。このように、SPSS Modeler がインストールされていない環境で使用するための完全な SPSS Modeler ストリームを公開して展開することができます。SPSS Modeler Solution Publisher は、個別のライセンスが必要とされている IBM SPSS Collaboration and Deployment Services - Scoring サービスの一部として配布されています。このライセンスを使用すると、SPSS Modeler Solution Publisher Runtime を受信し、公開されたストリームを実行することができます。

SPSS Modeler Solution Publisher について詳しくは、IBM SPSS Collaboration and Deployment Services の資料を参照してください。IBM SPSS Collaboration and Deployment Services Knowledge Center に『IBM SPSS Modeler Solution Publisher』と『IBM SPSS Analytics Toolkit』というセクションがあります。

## IBM SPSS Collaboration and Deployment Services 用の IBM SPSS Modeler Server アダプター

さまざまな IBM SPSS Collaboration and Deployment Services 用のアダプターを使用すると、SPSS Modeler および SPSS Modeler Server を IBM SPSS Collaboration and Deployment Services リポジトリとインタラクティブに機能させることができます。このように、リポジトリに展開された SPSS Modeler ストリームは、複数のユーザーで共有したり、シンクライアント アプリケーションである IBM SPSS Modeler Advantage からアクセスしたりできます。リポジトリをホストするシステムに、アダプターをインストールします。

## IBM SPSS Modeler のエディション

---

SPSS Modeler は次のエディションで使用できます。

### SPSS Modeler Professional

SPSS Modeler Professional は、CRM システムで追跡する行動や対話、人口統計データ、購入行動や販売データなど、多くの構造化データを処理するために必要なすべてのツールを提供しています。

### SPSS Modeler Premium

SPSS Modeler Premium は、特化したデータ、または構造化されていないテキスト・データを処理するために SPSS Modeler Professional を拡張する、別途ライセンスが必要な製品です。SPSS Modeler Premium には、以下の IBM SPSS Modeler Text Analytics が含まれます。



**IBM SPSS Modeler Text Analytics** は、高度な言語技術と Natural Language Processing (NLP) を使用して、構造化されていない多様なテキスト・データをすばやく処理し、重要なコンセプトを抽出および組織化し、そしてそのコンセプトをカテゴリー別に分類します。抽出されたコンセプトとカテゴリーを、人口統計のような既存の構造化データと組み合わせ、IBM SPSS Modeler の豊富なデータ・マイニング・ツールを適用する方法で、焦点を絞ったより良い決定を下すことができます。

### IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription は、従来の IBM SPSS Modeler クライアントとすべて同じ予測分析機能を提供します。Subscription エディションの場合、定期的に製品アップデートをダウンロードできます。

## 資料

資料は、SPSS Modeler の「ヘルプ」メニューから参照できます。ここからオンラインの Knowledge Center が開きます。Knowledge Center は、製品の外部で公に利用できます。

各製品の完全な資料(インストール手順を含む)は、PDF 形式でも提供されており、製品ダウンロードの一部として、個別の圧縮フォルダーに格納されています。また、最新の PDF 文書を Web サイト <https://www.ibm.com/support/pages/spss-modeler-1822-documentation> からダウンロードすることもできます。

## SPSS Modeler Professional ドキュメント

SPSS Modeler Professional のドキュメントスイート(インストール手順を除く)は次のとおりです。

- **IBM SPSS Modeler ユーザーズ・ガイド:** SPSS Modeler の使用への全体的な入門で、データ・ストリームの作成方法、欠損値の処理方法、CLEM 式の作成方法プロジェクトとレポートの処理方法と、IBM SPSS Collaboration and Deployment Services または IBM SPSS Modeler Advantage に展開するためのストリームのパッケージ方法が含まれています。
- **IBM SPSS Modeler 入力ノード、プロセス・ノード、出力ノード:** 各種形式のデータの読み取り、処理、および出力に使用するすべてのノードの説明です。これは、モデル作成ノード以外のすべてのノードについての説明です。
- **IBM SPSS Modeler モデル作成ノード:** データ・マイニング・モデルの作成に使用するすべてのノードについての説明です。IBM SPSS Modeler には、マシン学習、人工知能、および統計に基づいたさまざまなモデル作成方法が用意されています。
- **IBM SPSS Modeler アプリケーション・ガイド:** このガイドの例では、特定のモデル作成手法および技法について、簡単に対象を絞って紹介します。本ガイドのオンラインバージョンは、「ヘルプ」メニューからも利用できます。詳しくは、トピック 4 ページの『[アプリケーションの例](#)』を参照してください。
- **IBM SPSS Modeler Python スクリプトとオートメーション:** Python スクリプトによるシステムの自動化に関する情報です。ノードおよびストリームの操作に使用できるプロパティーを含めて説明します。
- **IBM SPSS Modeler 展開ガイド:** IBM SPSS Deployment Manager のもとで処理されるジョブ内のステップとして IBM SPSS Modeler ストリームを実行することに関する情報。
- **IBM SPSS Modeler CLEF 開発者ガイド:** CLEF では、IBM SPSS Modeler のノードとしてデータ処理ルーチンやモデル作成アルゴリズムなどのサード・パーティー製のプログラムを統合できます。
- **IBM SPSS Modeler データベース内 マイニング・ガイド:** サード・パーティー製アルゴリズムを使用してご使用のデータベースの能力を利用してパフォーマンスを向上させ、分析機能の範囲を拡張する方法に関する情報を示します。
- **IBM SPSS Modeler Server 管理およびパフォーマンス・ガイド:** IBM SPSS Modeler Server の構成方法と管理方法に関する情報。
- **IBM SPSS Deployment Manager ユーザー・ガイド:** IBM SPSS Modeler Server の監視や構成を行うための Deployment Manager アプリケーションに組み込まれている管理コンソール・ユーザー・インターフェースの使用法に関する情報。
- **IBM SPSS Modeler CRISP-DM ガイド:** SPSS Modeler でのデータ・マイニングに対する CRISP-DM 方法の使用に関するステップバイステップのガイドです。

- **IBM SPSS Modeler Batch ユーザーズ・ガイド:** IBM SPSS Modeler をバッチ・モードで使用するための完全ガイドで、バッチ・モードでの実行およびコマンド・ライン引数の詳細について説明します。このガイドは、PDF 形式のみです。

## SPSS Modeler Premium ドキュメント

SPSS Modeler Premium のドキュメントスイート (インストール手順を除く) は次のとおりです。

- **SPSS Modeler Text Analytics ユーザーズ・ガイド:** SPSS Modeler でテキスト分析を使用する場合の情報。テキスト・マイニング・ノード、インタラクティブ・ワークベンチ、テンプレートなどについて説明します。

## アプリケーションの例

SPSS Modeler のデータ・マイニング・ツールは、多様なビジネスおよび組織の問題解決を支援しますが、アプリケーションの例では、特定のモデル作成手法および技術に関する簡単に、目的に沿った説明を行います。ここで使用するデータ・セットは、データ・マイニング担当者が管理するような大規模データ・ストアと比較すると非常に小規模ですが、関係する概念および手法は実際のアプリケーションにも拡張できます。

その例を参照するには、SPSS Modeler の「ヘルプ」メニューから「**アプリケーションの例**」をクリックしてください。

データ・ファイルとサンプル・ストリームは、製品のインストール・ディレクトリーの **Demos** フォルダにインストールされています。詳しくは、4 ページの『**Demos フォルダ**』を参照してください。

**データベース・モデル作成の例:** 例は、『*IBM SPSS Modeler データベース内マイニング・ガイド*』を参照してください。

**スクリプトの例:** 例は、『*IBM SPSS Modeler スクリプトとオートメーションガイド*』を参照してください。

## Demos フォルダ

アプリケーションの例で使用されるデータ・ファイルとサンプル・ストリームは、製品のインストール・ディレクトリーの **Demos** フォルダ (例: `C:\Program Files\IBM\SPSS\Modeler\<version>\Demos`) にインストールされています。このフォルダには、Windows 「スタート」メニューの IBM SPSS Modeler プログラム グループから、または「**ファイル**」 > 「**ストリームを開く**」ダイアログ・ボックスの最近使ったディレクトリの一覧で「**Demos**」をクリックしてアクセスすることもできます。

## ライセンスの追跡

SPSS Modeler を使用すると、ライセンスの使用状況が一定の間隔で追跡され、ログに記録されます。ログに記録されるライセンス メトリックは `AUTHORIZED_USER` と `CONCURRENT_USER` であり、ログに記録されるメトリックのタイプは、SPSS Modeler に使用するライセンスのタイプによって決まります。

作成されたログ ファイルは IBM License Metric Tool によって処理可能であり、そのファイルからライセンス使用状況レポートを生成できます。

ライセンス ログ ファイルは、SPSS Modeler クライアント ログ ファイルが記録されるディレクトリと同じディレクトリに作成されます (デフォルトでは `%ALLUSERSPROFILE%\IBM\SPSS\Modeler/<version>/log`)。

## 第 2 章 IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher は、データ・マイニングの結果をビジネス・プロセスに組み込んで、実際の問題を解決するための、非常に役に立つツールです。IBM SPSS Modeler Solution Publisher を使用することで、外部のランタイム エンジンで実行したり、外部アプリケーションに埋め込むことができる「パッケージ」版のストリームを作成できます。これにより、データ・モデル作成ストリームを実稼働環境に展開して、日常的なビジネス・プロセスをサポートし、データのマイニングから取得した知識で組織の意思決定者を支援できます。

IBM SPSS Modeler Solution Publisher を使用すると、完全な *IBM SPSS Modeler* ストリームを公開し、展開できるため、単にモデルを (PMML として) エクスポートする以上の機能が提供されます。つまり、モデルに基づいた予測を作成する前に、データの集計、レコードの選択、または新しいフィールドの派生など、レコードおよびフィールド設定とともに、データの準備を実行することができます。その後、データを保存する前に、モデルの結果をさらに処理することができます。これらすべてを公開ストリームを実行することで実行します。

**注:** 公開は、該当するエクスポート・ノードの「公開」タブを使用して行います。詳しくは、[6 ページの『ストリームの公開』](#)のトピックを参照してください。バージョン 11.0 より前の製品のリリースに保存されているストリームでは、個別のパブリッシャー・ノードを使用していました。ノードは下位互換性に対応していますが、ノード・パレットでは使用できません。

### IBM SPSS Modeler Solution Publisher のライセンス交付

**注:** IBM SPSS Modeler Solution Publisher は、個別のライセンスが必要とされている IBM SPSS Collaboration and Deployment Services スコアリング・サービスの一部として配布されています。<http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

ライセンスを購入すると、公開済みストリームを実行できるようにする個別の IBM SPSS Modeler Solution Publisher Runtime が提供されます。詳しくは、[15 ページの『Runtime のインストール』](#)のトピックを参照してください。

### ロギング

ログ・ファイルは、IBM SPSS Modeler Solution Publisher インストール・ディレクトリーの config フォルダにある log4cxx.properties 構成ファイルにより制御されます。log4cxx.properties 構成ファイルを変更して、ロギング出力をカスタマイズできます。

Linux/UNIX を使用している場合、アプリケーションが構成ファイルを見つけられるように、アプリケーションを実行する前に環境変数を定義する必要があります (例えば、`export MODELERRUNTIME=MSP_INSTALLATION_DIRECTORY`)。

## IBM SPSS Modeler Solution Publisher の動作方法

IBM SPSS Modeler Solution Publisher を使用したソリューションの展開には、ストリームの公開とストリームのストリーム という 2 つのフェーズが含まれます。

**公開:** データ・マイニング・プロセスを進めると、最終的に、ビジネスの問題にすぐれた解決策を提供するモデルに到達します。この時点で、モデルを採用し、そのモデルをビジネス・プロセスに適用する準備ができています。ストリームを公開するときに、そのストリームの詳細な説明がディスクに (画像 ファイルおよびパラメーター・ファイルとして) 書き込まれます。詳しくは、[6 ページの『ストリームの公開』](#)のトピックを参照してください。

**実行:** ストリームの公開後は、公開されたストリームを実行して、ストリームに実装されているプロセスを再作成できます。この作業を行うには、スタンドアローンの IBM SPSS Modeler Runtime (*modelerrun.exe*) を使用するか、IBM SPSS Modeler Runtime Library を使用してストリームを実行するアプリケーションを開発します。(Runtime またはカスタム アプリケーションを使用して) IBM SPSS Modeler の外部でストリ

ームを実行するには、IBM SPSS Modeler Solution Publisher Runtime を最初にインストールする必要があります。詳しくは、[15 ページの『Runtime のインストール』](#)のトピックを参照してください。

## 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、IBM SPSS Modeler Solution Publisher Runtime または外部アプリケーションを使用して公開されたストリームが実行されるたびに、書き込まれる結果の形式が決定されます。例えば、発行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。

### ストリームの公開方法

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、公開されるファイルのルート名 (.pim、.par、および .xml の各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:** 公開される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。ストリームを再発行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、イメージやそのデータ・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。イメージを実行する場合、\*.par 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、公開されるイメージ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発行できます。
- **ストリームを公開します。** 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:**

- 発行済みのストリームを新規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、入力フィールド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と同じである必要があります。

- 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。





## 第3章 公開済みストリームの実行

公開済みのストリームを実行することの主要な効果は、常にデータ・セットを生成することです。そのデータは、ディスクファイルに保存するか、データベースに書き込むことができます。発行済みのストリームは、グラフ、テーブル、生成されたモデル、その他データ出力以外のものを生成できません。

IBM SPSS Modeler Solution Publisher を使用して公開されたストリームは、IBM SPSS Modeler Solution Publisher Runtime プログラムを使用して実行できます。Runtime プログラムである `modelerrun.exe` は、コマンド・ラインから開始します。画像ファイル(および任意で)実行するパラメーター・ファイルを表すオプションも指定します。Runtime コマンドは、次のとおりです。

```
modelerrun -nobanner -p <parameter-file> -o <options> <image-file>
```

このコマンド入力項目を次の表に示します。

| コマンド入力項目         | 説明   |
|------------------|--|
| -nobanner        | ランタイムのスタートアップ テキスト・メッセージを隠します(任意)。   |
| <parameter-file> | パラメーター・ファイルのファイル名を指定します(任意)。   |
| <options>        | Runtime の実行オプションを指定する option=value ペアのコンマ区切りリストです。オプションは、次に説明するとおりです(任意)。注: コンマの後には空白を入れないでください。 |
| <image-file>     | 実行する、画像ファイル(*.pim)のファイル名(必須)を指定します。  |

選択可能な実行オプションを次の表に示します。

| 名前                    | デフォルト値 | 説明   |
|-----------------------|--------|--|
| max_file_size         | -1     | 保存されるファイルの最大サイズ。値 -1 は制限がないことを示します。  |
| max_sql_string_length | 2,048  | データベースから SQL でインポートされる文字列の最大の長さ。これより長い文字列の値は、警告なしに右側で切り捨てられます。有効な範囲は 1 文字から 65,535 文字です。 |
| memory_usage          | 100    | 動的メモリー割り当てのための乗数。この値を上下に調整して、サーバーの総メモリー使用量を規制します。  |
| temp_directory        | "      | 一時ファイルに使用されるディレクトリーを指定します。   |
| request_passwords     | Y      | データベース・アクセスを必要とする公開済みのストリームを実行するときに、データベースのパスワードを求めるかどうか(Y/N)を制御します。                     |

注: ユーザー入力ノードを含むストリームを公開するときに、SPSS Modeler Solution Publisher でノード用の入力ファイルが必要なことに注意してください (SPSS Modeler の場合は、フィールドおよびそのデータをノード内で直接定義します)。デフォルトのファイル名は userInput.dat で、パラメーター・ファイル内で変更できます。

注: SPSS Modeler Solution Publisher で CPLEX の最適化ノードを含むストリームを実行している場合、デフォルトでは、組み込みの Community Edition CPLEX ライブラリーが使用されます。この場合、変数は 1000 個まで、制約も 1000 件までに制限されます。完全版の IBM ILOG CPLEX をインストールして、このような制限のない完全版の CPLEX エンジンに代わりに使用したい場合は、ご使用のプラットフォームに応じて以下の手順を実行してください。

- Windows の場合は、modelerrun.exe のコマンド・ライン引数として OPL ライブラリーのパスを追加します。例:

```
-o cplex_opl_lib_path="<CPLEX_path>%opl%bin%<Platform_dir>"
```

ここで、<CPLEX\_path> は CPLEX のインストール・ディレクトリー (C:\Program Files\IBM\ILOG\CPLEX\_Studio127 など) であり、<Platform\_dir> はプラットフォーム固有のディレクトリー (x64\_win64 など) です。

- Linux の場合は、modelerrun を編集して OPL ライブラリーのパスを追加します。例:

```
CPLEX_OPL_LIB_PATH=<CPLEX_path>/opl/bin/<Platform_dir>
```

ここで、<CPLEX\_path> は CPLEX のインストール・ディレクトリー (/root/Libs\_127\_FullEdition/Linux\_x86\_64 など) であり、<Platform\_dir> はプラットフォーム固有のディレクトリー (x86-64\_linux など) です。

### IBM SPSS Modeler Solution Publisher でのローカル Apache Spark ストリームの実行

IBM SPSS Modeler Solution Publisher でローカルの Spark ストリームを実行する場合、modelerrun コマンドで以下を指定する必要があります (C:/Anaconda2/python.exe にはご使用の Python 実行可能ファイルへのパスを指定します)。

```
-o eas_pyspark_python_path=C:/Anaconda2/python.exe
```

## パラメーター・ファイルを使用したストリーム実行のカスタマイズ

ストリームを公開するときに、画像ファイルには、データ・ソースや、元のストリームで選択されていた出力データの出力先などに関する情報が格納されています。しかし、多くの場合、他のデータ・ソースからストリームを実行することや、出力データを他の出力先に再ルートすることは役に立ちます。これは、ストリームのパラメーター・ファイルを編集する方法で実行できます。

ストリームを作成したのとは異なるコンピューターで、発行済みのストリームを実行する場合、このことは特に重要です。ターゲット・コンピューターの構造を反映するために、入力ファイルと出力ファイルの場所を更新する必要があります。公開済みのストリームが相対パスをサポートしていない場合、パラメーター・ファイル内でデータの場所を検証することは非常に重要です。(ストリームを公開するときにエクスポート・ノードのオプションで相対パスを指定した場合、IBM SPSS Modeler では公開済みストリームに合わせて画像ファイルとパラメーター・ファイルを作成する前に、その相対パスが絶対パスに変換されます。)

また、パラメーター・ファイルを使用して実行オプションを設定し、ストリーム・パラメーターの値を変更することができます。

パラメーターの値は、パラメーター・ファイル内で <name.attribute>=<value> ペアとして指定されます。特定のパラメーターでは、パラメーター名の中で、ストリーム内にある同じタイプのノードを識別するための ID の数字が使用されています。例えば、入力と出力の両方が特定のデータベース接続を参照しているストリームでは、入力接続について記述しているパラメーターは dbconn0.datasource、dbconn0.user などであり、出力接続に関連付けられているパラメーターは dbconn1.datasource、dbconn1.user などということがあります。値は必ず、二重引用符で囲まれています。パラメーター・フ



ファイルには、次のパラメーターが含まれています。これらすべてを変更して、ストリームの実行をカスタマイズできます。

| 表 3. 実行パラメーター                |  |
|------------------------------|--|
| パラメーター                       | 目的   |
| dbconnx.datasource           | データ・ソース名 (DSN)。  |
| dbconnx.user                 | 制限された方法でデータベースにアクセスするためのユーザー名。   |
| dbconnx.password             | 制限された方法でデータベースにアクセスするためのパスワード。   |
| dbconnx.epassword            | 制限された方法でデータベースにアクセスするための暗号化パスワード。暗号化されたパスワードを生成するには、IBM SPSS Modeler ユーザー・インターフェースの「ツール」メニューから「パスワードの暗号化」を選択します。暗号化パスワードは、必要に応じてコピーして貼り付けます。 |
| dbconnx.password_required    | このデータベース接続で使用するパスワードの入力を求めるかどうかを決定するフラグ。(値は、"Y" または "N" です。)公開時にパスワードが指定されていない場合、自動的に "N" に設定されます。   |
| dbconnx.catalog              | データベース接続に関連付けられるカタログです。  |
| dbtablex.name                | ベース・テーブル名。注:新しいテーブル内にあるフィールドの順序は、最初に公開されたストリーム内にあったフィールドの順序と同じであることが必要です。  |
| dbtablex.schema              | スキーマ名。   |
| dbtablex.catalog             | 現在のテーブルに関連付けられるカタログです。   |
| filex.name                   | データ・ファイルのファイル名です。注:新しい入力ファイル内にあるフィールドの順序は、最初に公開されたストリーム内にあったフィールドの順序と同じであることが必要です。   |
| filex.path                   | ファイルの場所 (ディレクトリー名)。  |
| filex.field_names_included   | テキスト形式のファイルでは、このフラグは、フィールド名がファイルの最初の行として記述されているかどうかを示します。(値は、"Y" または "N" です。)  |
| filex.field_separator        | テキスト形式のファイルでは、フィールド値を区切るために使用される文字です。  |
| filex.decimal_separator      | テキスト形式のファイルでは、小数点を表すために使用される文字です。  |
| options.angle_in_radians     | CLEM 式で使用する三角関数の角度単位として、「ラジアン」を使用するかどうかを示すフラグです。(値は、"Y" または "N" です。)   |
| options.date_2digit_baseline | 日付の年を 2 桁で表記する場合に、どの世紀を使用するのか定義します。  |
| options.date_baseline        | CLEM 日付関数で単一日付を扱う場合に、使用する基準の年 (月日は常に 1 月 1 日) を指定します。  |
| options.time_rollover        | 時間の差が負である場合に、それが過去を表すかどうかを示すフラグです。   |

| 表 3. 実行パラメーター (続き)        |   |
|---------------------------|---|
| パラメーター                    | 目的  |
| options.decimal_separator | テキスト・ファイルで、小数点を表すデフォルトの文字です。  |
| options.time_format       | 文字列が CLEM 時間関数によって時間として解釈された場合に使用する時間の形式です。   |
| options.date_format       | 文字列が CLEM 日付関数によって日付として解釈された場合に使用する日付の形式です。   |
| options.timestamp_format  | タイムスタンプ・フィールドが ODBC データ・ソースによって文字列として読み取られる場合に使用される形式です。  |
| paramx.value              | ストリーム・パラメーター値。<br><br>注: パラメーター名、説明、およびストレージ・タイプが参照用に含まれますが、変更することはできません。値はそのタイプに対して有効である必要があります。 |

## IBM SPSS Modeler Solution Publisher のアプリケーションへの埋め込み

IBM SPSS Modeler Solution Publisher は、スタンドアローンのランタイム エンジンとして使用する以外に、ランタイム・プログラミング・ライブラリー (CLEMRTL) を提供することもできます。他のプログラムはこれをおして、IBM SPSS Modeler Solution Publisher のストリーム実行を制御できます。C および C++ で記述されたクライアント プログラムの CLEMRTL を呼び出すことができます。CLEMRTL を使用するには、ヘッダー・ファイル *clemrtl.h* (`..¥installation¥clemrtl¥include` フォルダーで利用可能) を含め、アプリケーションの作成時に、開発プラットフォームに適したライブラリー・ファイルをそのアプリケーションにリンクする必要があります。clemrtl.dll ファイルは、`..¥installation¥bin` フォルダーから使用できます。

ライブラリー・手続きを参照するソース・ファイルには、ヘッダー *clemrtl.h* を含める必要があります。このヘッダー・ファイルは、ライブラリー・手続きに対する ANSIC のプロトタイプを提供し、役立つマクロを定義します。使用中のプログラムが必要とするもの以外に、他のヘッダーを含める必要はありません。名前の競合を防ぐため、すべてのライブラリー関数名は `clemrtl_` で始まり、すべてのマクロ名には `CLEMRTL_` という接頭辞が付きます。

CLEMRTL 手続きは、役立つ情報を報告します。その情報は、何らかの方法でアプリケーションに返す必要があります。CLEMRTL は、そのようなメッセージを扱うためのメカニズムを用意しています。

- アプリケーションでは、`clemrtl_getErrorDetail()` 関数を使用して最後のエラーの詳細を取り出せます。
- アプリケーションでは、`clemrtl_setReportHandler()` 関数を使用して独自のレポート処理手続きを提供できます。

Runtime システムには、レポートで使用できるローカライズされたメッセージ カタログがあります。アプリケーションに渡されるどのレポートにも、ローカライズされたメッセージ文字列が含まれています。レポートには、レポート・コードも含まれているので、アプリケーションはそのメッセージを解釈して異なる方法で表示することもできます。詳細な (ローカライズされた) メッセージを取得するには、アプリケーションの *config* ディレクトリーに適切な *messages.cfg* ファイルが保存されている必要があります。異なるロケールのさまざまなバージョンの *messages.cfg* が `<installdir>/config/<locale>` のディレクトリーにあります。 `<installdir>` は IBM SPSS Modeler Solution Publisher をインストールしたディレクトリー、 `<locale>` は使用するロケールです。 *messages.cfg* の適切なバージョンを選択して、それをプロジェクト・ディレクトリーの *config* サブディレクトリーに追加します。

Runtime システムを正しく動作させるために、特別な環境やレジストリ設定は必要ありません。従属ライブラリーは、CLEMRTL にリンクされたアプリケーションとともに配布する必要があります。これらのファイルは、ディレクトリー `<installdir>/bin` に追加されます。

### 要件

ランタイム・プログラミング・ライブラリーでテスト済みのプラットフォームおよびコンパイラーを、次の表に示します。

| 表 4. プラットフォームとコンパイラ |  |
|---------------------|--|
| プラットフォーム            | 推奨するコンパイラ                                    |
| AIX                 | IBM XL C/C++ Enterprise Edition V8.0 for AIX |
| Linux 64 ビット        | gcc 3.2.3、Intel C++ compiler 9.1.043         |
| Windows             | Visual Studio 2008 R2、Visual Studio 2015     |

## IBM SPSS Modeler Solution Publisher サンプル・アプリケーション

アプリケーションの例 `dlltest.c` が IBM SPSS Modeler Solution Publisher インストール・ディレクトリーの `¥clemtrl¥demo` フォルダーに収められています。アプリケーションの例は、独自のアプリケーションにランタイム ライブラリーを埋め込むためのガイダンスとして使用できます。

例のアプリケーションで使用される情報は `¥clemtrl¥demo` フォルダー内にあります。詳細は、`README.txt` ファイルを参照してください。アプリケーションのコンパイル方法はプラットフォームによって異なるため、作成ファイル例も用意されています。



# 第 4 章 IBM SPSS Modeler Solution Publisher Runtime のインストール

## Runtime のインストール

IBM SPSS Modeler Solution Publisher Runtime をインストールすると、公開されたストリームを実行することができます。公開されたストリームはオペレーティング・システムには依存していないため、ランタイムがサポートしている任意のプラットフォーム上で、公開されたストリームを実行することができます。

IBM SPSS Modeler からストリームを公開できるようにする機能は、IBM SPSS Modeler とともにインストールされます。注意を要するのは、IBM SPSS Modeler のリリースのたびに、公開済みファイルを改めてエクスポートする必要があることです。例えば、以前のバージョンを使用してエクスポートされたファイルは、現行バージョンの IBM SPSS Modeler Solution Publisher Runtime では動作しません。

## Windows へのインストール

次の項で、SPSS Modeler Solution Publisher Runtime を Windows オペレーティング・システムにインストールするための、手順を説明します。

1. IBM SPSS Modeler Solution Publisher のインストールを開始します。
2. インストール・ウィザードで、「次へ」をクリックして開始します。
3. 画面に表示される指示に従います。「次へ」をクリックして作業を続行します。
4. すべてのオプションを設定したら、インストールの準備が完了します。「インストール」をクリックすると、ファイルの転送が開始されます。
5. すべてのファイルがインストールされたら、「完了」をクリックします。

## UNIX へのインストール

次の項で、IBM SPSS Modeler Solution Publisher Runtime を Unix オペレーティング・システムにインストールするための、追加のシステム要件と手順を説明します。

### UNIX の追加要件

システムのカーネルの制限が、IBM SPSS Modeler Server の操作に対して十分であることを確認する必要があります。データ、メモリー、ファイル、およびプロセスの `ulimit` 設定は特に重要です。IBM SPSS Modeler Server 環境では無制限に設定してください。これを行うには、次のようになります。

1. 次のコマンドを `modelersrv.sh` に追加します。

```
ulimit -d unlimited
```

```
ulimit -m unlimited
```

```
ulimit -f unlimited
```

```
ulimit -u unlimited
```

また、スタック制限をシステムで許可される最大値に設定します (`ulimit -s XXXX`)。例:

```
ulimit -s 64000
```

2. IBM SPSS Modeler Server を再起動します。

インストーラーがインストール・ファイルを解凍できるようにするには、`gzip` ファイル圧縮ユーティリティー、および `GNU cpio` がインストールされていて、`PATH` 環境変数に登録されているパスに存在すること

も必要です。さらに、SPSS Modeler Server を実行しているマシンで、ロケールを EN\_US.UTF-8 に設定する必要があります。

## UNIX インストール手順

ここでは、IBM SPSS Modeler Solution Publisher Runtime for Unix のインストール手順を説明しています。インストールを行う際には、インストール先ディレクトリーに対する読み込み権と書き込み権が必要になります。適切なアクセス権を持つアカウントでログ・オンしてください。

注：インストールするには `root` としてログインする必要があります。

1. インストールパッケージの `modelrun` ディレクトリーを開きます。
2. 関連するプラットフォーム・ディレクトリーに変更します。
3. `.bin` インストール・スクリプトを実行します。例:

```
./modelersolutionpublisherhpia64.bin -i console
```

4. 導入の詳細が表示されます。続行するには、Enter キーを押してください。
5. ライセンス情報が表示されます。ライセンスを読み、承認するには `1` を入力し、Enter を押して続行します。
6. 使用できる言語のリストが表示されます。インストールする言語の番号を入力し、Enter キーを押して続行します。
7. インストール場所を入力する指示が表示されます。デフォルト・ディレクトリーの `/usr/IBM/SPSS/ModelerSolutionPublisher<nn>` (`<nn>` はバージョン番号) を使用する場合は、Enter キーを押してください。指定したディレクトリーにランタイムがインストールされます。
8. インストール場所を確認する指示が表示されます。正しければ `y` を入力して Enter キーを押します。
9. そこまでの入力項目を確認するために、プリインストールの要約が表示されます。続行するには、Enter キーを押してください。
10. インストールのルーチンを実行する準備ができたことを示すメッセージが表示されます。続行するには、Enter キーを押してください。
11. インストールのルーチンが実行される間、進捗状況バーが表示されます。インストールが完了したら、Enter キーを押してインストーラーを終了します。

## UNIX での ODBC の構成

### IBM SPSS Modeler Solution Publisher Runtime とともに起動するように ODBC を構成する手順

IBM SPSS Modeler Server からデータベースに接続できる場合、IBM SPSS Modeler Solution Publisher Runtime の起動スクリプトから同じ `odbc.sh` スクリプトを参照して、同じサーバーに IBM SPSS Modeler Solution Publisher Runtime のインストールを設定することができます。

1. IBM SPSS Modeler Solution Publisher Runtime で `modelerrun` スクリプト・ファイルを編集して、スクリプトの最後の行のすぐ上に次の行を追加します。

```
. <odbc.sh_path>
```

ここで、`odbc.sh_path` は `odbc.sh` ファイルの絶対パスです。例:

```
. /usr/spss/odbc/odbc.sh
```

注：ここではシンタックスが重要になります。最初のピリオドとファイルへのパスの間に必ずスペースを入れてください。

2. `modelerrun` スクリプト・ファイルを保存します。
3. デフォルトでは、DataDirect Driver Manager は UNIX システムで ODBC を使用するように IBM SPSS Modeler Solution Publisher Runtime 向けに構成されていません。DataDirect Driver Manager を読み込

むように UNIX を構成するには、以下のコマンドを入力します (ここで、sp\_install\_dir は Solution Publisher Runtime のインストール ディレクトリーです)。

```
cd sp_install_dir
rm -f libspssodbc.so
ln -s libspssodbc_datadirect.so libspssodbc.so
```

## インストールのトラブルシューティング

### インストール時の無効なデジタル署名

IBM SPSS Modeler 製品は、デジタル署名に IBM 発行の証明書を使用します。特定の環境で、SPSS Modeler 製品をしようとする際に次のエラーが発生する場合があります。

```
Error 1330. A file that is required cannot be installed because the cabinet file filename has an invalid digital signature...
```

### すべての Windows ユーザー

インターネットに接続しておらず、正しい証明書がインストールされていないマシン上で SPSS Modeler 製品をインストールしようとすると、このメッセージが表示されます。この問題を解決するには、次の手順に従います。

1. 「OK」をクリックして、メッセージに応答します。
2. 「キャンセル」をクリックしてインストーラーを終了します。
3. インストールするマシンがインターネットに接続していない場合、インターネット接続マシンで次のステップを実行し、.cer ファイルをインストールするマシンにコピーします。
4. <https://support.symantec.com> にアクセスし、「VeriSign Class 3 Primary Certification Authority - G5 ルート証明書」を検索して、ダウンロードします。それを .cer ファイルとして保存します。
5. .cer ファイルをダブルクリックします。
6. 「全般」タブで、「証明書のインストール」をクリックします。
7. 証明書インポート・ウィザードの指示に従い、デフォルトのオプションを選択して最後に「完了」をクリックします。
8. インストールを再実行します。

## Runtime の起動

Runtime をインストールしたら、それを使い、エクスポート・ノードを使用して IBM SPSS Modeler から公開されたストリームを実行することができます。詳しくは、6 ページの『ストリームの公開』のトピックを参照してください。公開された画像ファイル名や他の実行パラメーター・オプションを指定して、コマンド・ラインからランタイムを開始します。

ランタイムを実行するには、コマンド・プロンプトに次のコマンドを入力します。

```
modelerrun -p <parameter-file> -o <options> <image-file>
```

このコマンド入力項目を次の表に示します。

| コマンド入力項目         | 説明                                    |
|------------------|---------------------------------------|
| <parameter-file> | 公開されたパラメーター・ファイルのファイル名を指定します (オプション)。 |

| 表 5. Runtime コマンド入力項目 (続き) |   |
|----------------------------|---|
| <options>                  | Runtime の実行オプションを指定する option=value ペアのコンマ区切りリストです。注: コンマの後には空白を入れないでください。 |
| <image file>               | 実行する、公開された画像ファイル (*.pim) のファイル名を指定します。                                    |

注: UNIX 環境で中国語または日本語などマルチバイトの言語を使用する場合、使用する言語コードを指定する必要があります。次の例は、簡体字中国語の言語コードを示します。

```
./modelerrun -o locale="zh_CN.GB18030",encoding="GB18030" -p
```

ここで、-p は pim または par ファイルの場所を示します。

詳しくは、9 ページの『[第 3 章 公開済みストリームの実行](#)』を参照してください。

## 一時ディレクトリの変更

IBM SPSS Modeler Solution Publisher Runtime が行う処理や操作の中には、一時ファイルを作成する必要があるものもあります。Runtime のデフォルトでは、システムの一時ファイル用ディレクトリに一時ファイルが作成されます。一時ディレクトリの場所を変更するには、次の手順に従ってください。

注: 正しいシンタックスを使用することは非常に重要で、特にスペースとディレクトリ・パスには注意が必要です。

公開済みファイル *webtest.pim* および *webtest.par* とともに IBM SPSS Modeler Solution Publisher Runtime を使用するとします。この場合、一時ファイルの場所を変更して *C:¥¥published¥¥temp* を使用するようにします。実行可能ファイルが使用する一時ディレクトリの場所を変更するには、次のようにします。

1. *.pim* と *.par* のファイルが保存されているディレクトリから、次のコマンドを実行します。

```
modelerrun -o temp_directory="C:¥¥published¥¥temp" -p webtest.par webtest.pim
```

これで、Runtime によって、その特定プロセス用のみの一時ディレクトリとして、"*C:¥¥published¥¥temp*" が使用されるようになります。

## IBM SPSS Modeler Solution Publisher Runtime のアンインストール

次の項で、IBM SPSS Modeler Solution Publisher Runtime のアンインストール手順を説明します。

### Windows のアンインストール手順

1. Windows のコントロール・パネルから「アプリケーションの追加と削除」を開きます。
2. リストから、「IBM SPSS Modeler Solution Publisher <nn>」を選択します。
3. 「削除」をクリックします。
4. 画面に表示される指示にしたがって、ランタイムをアンインストールします。

注: プログラムを削除するには、管理者権限でコンピューターにログインする必要があります。

### UNIX のアンインストール手順

インストール・ファイルをすべて含めて、インストール・ディレクトリを削除します。



## 第 5 章 IBM SPSS Embedded Predictive Modeling API

IBM SPSS Embedded Predictive Modeling API で定義されているシンプルな Java インターフェースを使用して、IBM SPSS Modeler で設計した成果物をご使用のアプリケーションと組み込み方式で統合できます。ライブラリーを使用するコーダー向けの Javadoc 形式の完全な資料が、IBM SPSS Modeler Solution Publisher に付属しています。

Embedded Predictive Modeling (EPM) API を使用する際は、以下の制限事項に注意してください。

- **固有の入力ノード・ラベルとターミナル・ノード・ラベルが必要です。** EPM API はノード・ラベルを使用します。ラベルは常に入力ノードまたはターミナル・ノードのコンテキスト内にありますが、あいまいになる場合があります。
- **スーパーノードは入力ノードとしてもターミナル・ノードとしてもサポートされません。** スーパーノードにすべての操作で共通のソースを提供するために、スーパーノードは入力ノードとしてもターミナル・ノードとしてもサポートされません。
- **入力フィールド名の制限。** フィールド名には一部の文字を使用できません。

API 機能の概要として以下の情報を参照してください。詳しくは、製品に付属の Javadoc を参照してください。

### 予測モデルの組み込みスコアリング

**使用法:** 新規データを受け取ったときにオンデマンドで予測分析を生成するためにアプリケーションによって使用されます。アプリケーションは更新されたモデルを処理する必要があります。このスコアリング・モードでは、すべての入力ノード (およびスコアリング・ブランチのターミナル・ノード) がアプリケーションの入力および出力で置換されます。

**アクション:** アプリケーションによって管理された入力データから迅速かつ効率的に予測分析を生成する繰り返しの呼び出しのために、以前 IBM SPSS Modeler ストリームで設計したスコアリング・ブランチを準備します。

### 予測モデルの更新

**使用法:** ストリーム設計で履歴データを使用してモデル・アルゴリズムの初期学習を実行します。また、構成済みの予測モデル・アルゴリズムを最新の履歴データで定期的に「更新」します。

**アクション:** IBM SPSS Modeler ストリーム内のすべての予測モデル・ビルダー・ノードを実行します。これにより、データ・マイニング手法を使用した構成に従って予測モデル・アルゴリズムの学習が行われます。各モデル・ビルダー・ノードにリンクされているモデル・アプライヤー・ナゲットを更新します。ストリーム・パラメーターおよび入力ノード/ターミナル・ノードのプロパティを使用して、ある程度の力学を実装できます。

### 予測モデルの評価

**使用法:** IBM SPSS Modeler ストリームでの設計に従って学習した予測モデルの精度、信頼度、またはその他の質的評価を伝達するために、アプリケーションによって使用されます。この情報は、通常、アプリケーションで新たに更新された予測モデルのスコアリングへのプロモートを制御するために使用されます。モデル更新が新規 `PredictiveModel` オブジェクトを返す主な理由は、新たに更新されたモデルの評価データを元のデータと比較できるようにするためです。

**アクション:** IBM SPSS Modeler ストリーム内のすべてのドキュメント・ビルダー・ノード (主に IBM SPSS Modeler 出力パレットおよびグラフ・パレット上の非対話式出力ノード) を実行し、その結果生成されるドキュメントを返します。ストリーム・パラメーターおよび入力ノード/ターミナル・ノードのプロパティを使用して、ある程度の力学を実装できます。

## 予測モデルのストリーム実行

**使用法:** 「更新」、「評価」または2つのスコアリング・モードのいずれにも適合しないユース・ケースはすべて、ストリーム実行を使用して処理する必要があります。

**アクション:** 実行は、IBM SPSS Modeler ストリーム・オプションによって制御されます。「すべてのターミナルノード」モード (IBM SPSS Modeler クライアントで緑の大きい矢印ボタンをクリックするのと同様)、またはストリーム・レベルのスクリプトのコードに従って実行されます。ストリーム・パラメーターおよび入力ノード/ターミナル・ノードのプロパティを使用して、ある程度の力学を実装できます。

## 予測モデルの外部データのスコアリング

**使用法:** 入力データの完全なバッチが処理されるため、一般的に「バッチ」処理とも呼ばれます。

**アクション:** 設計に従ってスコアリング・ブランチを実行します。構成済みデータ入力ノードおよび生成された予測分析からの入力、ターミナル・ノードによる制御に従って維持されます。ストリーム・パラメーターおよび入力ノード/ターミナル・ノードのプロパティを使用して、ある程度の力学を実装できます。

## 予測モデルのストリーム・パラメーターと、入力ノードおよびターミナル・ノードのプロパティの仕様

**使用法:** よく行われる方法は、ストリームのブランチの式で参照されるパラメーターを定義することです。制御のもう1つのポイントは、入力ノードまたはターミナル・ノードの特定の主要プロパティを変更して、使用されているファイル・パスなどを変更することです。

**アクション:** IBM SPSS Modeler のストリーム実行の一部の局面は、アプリケーションによって動的に変更される場合があります。

**注:** パラメーターの変更は PredictiveScorer の作成前に行う必要があります (組み込みスコアリングに関する上述の説明を参照)。入力ノードおよびターミナル・ノードのプロパティは、このオブジェクトに対しては意味がありません。

# 第 6 章 IBM SPSS Modeler Solution Publisher Runtime Library API リファレンス

## はじめに

CLEMRTL API を使用すると、アプリケーションからの公開ストリームの実行を制御できます。この項では、使用可能な API 関数について説明します。

API を使用する上での一般的な問題を次に示します。

- API には最大互換性のための C-linkage がありますが、ライブラリーには C++ の依存関係があります。一部のプラットフォームの場合、このことは、API が C++-aware リンカーでのみ使用可能であることを意味します。
- ライブラリー内のいずれの関数を使用するよりも前に、`cleomrtl_initialize_ext()` を呼び出す必要があります。
- `cleomrtl_image_handle` のタイプがイメージ識別子として使用されます。
- すべての関数が 1 つのステータス指標コードを返します。ステータス指標の値を、次の表に示します。

| 結果            | 説明   |
|---------------|--|
| CLEMRTL_OK    | 成功   |
| CLEMRTL_FAIL  | エラー（追加情報はありません）  |
| CLEMRTL_ERROR | エラーの詳細については、 <code>cleomrtl_getErrorDetail()</code> 関数を使用して参照可能。 |

## API プロセスの概要

API を使用するアプリケーションの一般概要は次の通りです。

1. `cleomrtl_initialize_ext()` を使用してライブラリーを初期化します。
2. `cleomrtl_getOption()` と `cleomrtl_setOption()` を使用して、実行オプションの検査と変更を行います。
3. `cleomrtl_openImage()` を使用するイメージを開いて、イメージ ハンドルを受け取ります。
4. `cleomrtl_enumerateParameters()`、`cleomrtl_getParameter()`、および `cleomrtl_setParameter()` を使用して、イメージ・パラメーターを確認し、変更します。
5. `cleomrtl_execute()` を使用するイメージを実行します。同じイメージがパラメーターの変更なしに複数回実行される場合は、`cleomrtl_prepare()` を最初に使用します。
6. `cleomrtl_closeImage()` を使用するイメージを閉じます。
7. 進行中の実行操作をキャンセルするには、`cleomrtl_interrupt()` を使用します。
8. 最後のエラーについての情報を取得するには、`cleomrtl_getErrorDetail()` を使用します。到達したエラー・メッセージと他の診断メッセージを受け取るには、`cleomrtl_setReportHandler()` を使用します。

ライブラリーは、次の条件下でマルチスレッド使用できるように設計されています。

- イメージ ハンドルを引数として取らない複数のグローバル関数は、同時にコールすることはできません。

- イメージハンドルを引数として取るイメージ・ローカル関数は、同じイメージハンドルに適用した場合、同時にコールすることはできませんが、`clemrtl_interrupt()` は例外です。
- イメージ・ローカル関数は、異なるイメージハンドルに適用した場合は同時にコールできます。

このことは、特に、別々のスレッドが異なるイメージハンドルを使用するなら、イメージを同時に準備し実行できることを意味します。異なるイメージハンドルが同じイメージを参照する場合があります (`clemrtl_openImage()` は、同じイメージの場合でさえ、コールされるたびに新しいハンドルを生成します)。しかし、同じイメージの複数インスタンスを実行する際には、必要に応じて、イメージパラメータを変更して出力の方向変更を行うことを確実にしてください。同時実行の2つのイメージで、両方とも同じ出力のファイルまたはデータベース・テーブルに書き込む場合、結果は予測できないものになります。

## API 関数

次は、API で表示されるすべての関数の一覧です。

### initialise

この関数は以下の関数と同等です。

```
initialise_ext(flags, 0, 0);
```

新しいプログラムは `initialise_ext` をコールする必要があります。

### initialise\_ext

```
int clemrtl_initialise_ext(
    unsigned flags,
    int arg_count,
    const clemrtl_init_arg* args);
```

IBM SPSS Modeler Solution Publisher Runtime を初期化します。この関数は、他の API 関数の前にコールする必要があります。

| パラメーター                 | 説明  |
|------------------------|---|
| <code>flags</code>     | 初期化プロセスのいくつかのアスペクトを制御します。値は、以下で説明するフラグのビット単位 OR として構成されます。  |
| <code>arg_count</code> | 追加の初期化引数の数。args 配列の長さ以下である必要があります。  |
| <code>args</code>      | 追加の初期化引数。このタイプによって記述されている引数は次のとおりです。 <pre>typedef struct _clemrtl_init_arg {     const char* name;     const char* value; } clemrtl_init_arg;</pre> |

`name` フィールドは構成プロパティの名前、`value` フィールドはその値です。認識される構成プロパティを次の表に示します。

| 表 8. 構成プロパティ           |   |
|------------------------|---|
| 名前                     | 値   |
| installation_directory | IBM SPSS Modeler Solution Publisher インストール・フォルダーへのフル・パス。インストール・フォルダーには、bin フォルダ—および config フォルダ—が含まれています。 |

次の表にあるエラー・コードのいずれかを返します。

| 表 9. エラー・コード |                    |
|--------------|--------------------|
| 結果           | 説明                 |
| CLEMRTL_OK   | 成功。                |
| CLEMRTL_FAIL | 初期化エラー（追加情報はありません） |

flags パラメーターで、初期化プロセスの特定アスペクトを制御できます。0（または、CLEMRTL\_INIT\_DEFAULTS）の値は、以下のデフォルト処理を指定します。

- tzset() をコールしてデフォルトの時間区域を設定します。
- LC\_NUMERIC ロケールを「C」に設定します。
- メモリーを使い果たしたときに、グローバルで新しいハンドラを設定して例外を投げます。

この動作は、次の表に示すフラグの組み合わせを指定することによって変更できます。

| 表 10. フラグ                      |                                   |
|--------------------------------|-----------------------------------|
| フラグ                            | 説明                                |
| CLEMRTL_INIT_NO_TZ             | タイム・ゾーンを設定しません。                   |
| CLEMRTL_INIT_NO_LOCALE         | ロケールを変更しません。                      |
| CLEMRTL_INIT_NO_NEW_HANDLER    | 新しいハンドラを設定しません。                   |
| CLEMRTL_INIT_LOCAL_NEW_HANDLER | それぞれの API コール内で新しいハンドラをローカライズします。 |

ライブラリーが数値と文字列との間の変換を適切に行うためには、LC\_NUMERIC ロケールを「C」に設定する必要があります (IBM SPSS Modeler の数値の形式は、「C」ロケールのルールを使用するため)。これには、テキスト・ファイルで読み出しと書き込みが行われる数値の変換が含まれます。

- アプリケーションが LC\_NUMERIC ロケールに依存し、数値と文字列間での変換が必要ない場合は、NO\_LOCALE を指定します。
- NO\_NEW\_HANDLER と LOCAL\_NEW\_HANDLER は、互いに排反します。ライブラリーでは、メモリーを使い果たしたときに、new が例外を投げる必要があります。
- アプリケーションで独自の新しいハンドラが設定されるか、または、コンパイラ ランタイム システムがデフォルトで例外を投げるのが確実な場合は、NO\_NEW\_HANDLER を指定します。

メモリーを使い果たしたときにヌル値ポインタを返す new にアプリケーションが依存する場合にのみ、LOCAL\_NEW\_HANDLER を指定します。注：これをマルチスレッド環境で使用するには問題があります。

### getOption

```
int clemrtl_getOption(
    const char* name,
    char* value,
    int value_size);
```

実行オプションの値を取得します。

| 表 11. <i>getOption</i> パラメーター |                   |
|-------------------------------|-------------------|
| パラメーター                        | 説明                |
| name                          | オプション名。           |
| value                         | オプション値を受け取るバッファー。 |
| value_size                    | 値バッファーのサイズ。       |

次の表にあるステータス・コードのいずれかを返します。

| 表 12. ステータス・コード |                  |
|-----------------|------------------|
| 結果              | 説明               |
| CLEMRTL_OK      | 成功。              |
| CLEMRTL_FAIL    | エラー（追加情報はありません）。 |
| CLEMRTL_ERROR   | エラー（追加情報があります）。  |

オプション名が上記の実行オプション名のいずれかではない (`memory_usage` など) 場合、または値バッファーが値 (最後のヌル文字を含む) を受け取るのに十分な大きさではない場合、操作は失敗します。オプション値は常に、つまり解釈が数値で行われる場合でさえ、文字列で構成されます。詳しくは、[9 ページの『第 3 章 公開済みストリームの実行』](#)を参照してください。

### setOption

```
int clemrtl_setOption(
    const char* name,
    const char* value);
```

実行オプションの値を設定します。

| 表 13. <i>setOption</i> パラメーター |         |
|-------------------------------|---------|
| パラメーター                        | 説明      |
| name                          | オプション名。 |
| value                         | オプション値。 |

次の表にあるステータス・コードのいずれかを返します。

| 表 14. ステータス・コード |                  |
|-----------------|------------------|
| 結果              | 説明               |
| CLEMRTL_OK      | 成功。              |
| CLEMRTL_FAIL    | エラー（追加情報はありません）。 |
| CLEMRTL_ERROR   | エラー（追加情報があります）。  |

オプション名が上記の実行オプション名のいずれかではない (`memory_usage` など) 場合、操作は失敗します。オプション値は常に、つまり解釈が数値で行われる場合でさえ、文字列で構成されます。

### openImage

```
int clemrtl_openImage(
    const char* image_file,
    const char* param_file,
    clemrtl_image_handle* handle);
```

公開ストリーム イメージを開きます。

| パラメーター     | 説明                |
|------------|-------------------|
| image_file | 画像ファイル名。          |
| param_file | パラメーター・ファイル名。     |
| handle     | イメージ ハンドルを受け取ります。 |

次の表にあるステータス・コードのいずれかを返します。

| 結果            | 説明               |
|---------------|------------------|
| CLEMRTL_OK    | 成功。              |
| CLEMRTL_FAIL  | エラー（追加情報はありません）。 |
| CLEMRTL_ERROR | エラー（追加情報があります）。  |

別個のパラメーター・ファイルが要求されない場合、パラメーター・ファイルは NULL になることがあります。画像ファイルが指定されなかったり、どのファイル名も無効だったり、または、内容が読めなかったりする場合、操作は失敗します。操作が成功した場合、返されたイメージ ハンドルを使用して、後続の API コールのイメージ インスタンスを識別できます。

### closeImage

```
int clemrtl_closeImage(  
    clemrtl_image_handle handle);
```

イメージ ハンドルを閉じて、リソースを開放します。

| パラメーター | 説明         |
|--------|------------|
| handle | イメージ ハンドル。 |

次の表にあるステータス・コードのいずれかを返します。

| 結果            | 説明               |
|---------------|------------------|
| CLEMRTL_OK    | 成功。              |
| CLEMRTL_FAIL  | エラー（追加情報はありません）。 |
| CLEMRTL_ERROR | エラー（追加情報があります）。  |

イメージでもう 1 つの API コールが進行中の場合、特にそのイメージが実行中の場合、操作は失敗します。操作が成功した場合、イメージは閉じられ、影響を及ぼさないもう 1 つのコール `clemrtl_closeImage()` を除いて、ハンドルはいかなる後続の API コールにも使用することはできません。

### enumerateParameters

```
int clemrtl_enumerateParameters(  
    clemrtl_image_handle handle,  
    clemrtl_parameter_proc proc,  
    void* data);
```

それぞれのイメージパラメーターと値にコールバックの手順を適用します。

| 表 19. <i>enumerateParameters</i> パラメーター |                     |
|---|---------------------|
| パラメーター                                  | 説明                  |
| handle                                  | イメージ ハンドル。          |
| proc                                    | パラメーター・コールバックの手順    |
| data                                    | コールバックのためのユーザー指定データ |

次の表にあるステータス・コードのいずれかを返します。

| 表 20. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

コールバックの手順には次のような種類があります。

```
typedef void (*clemrtl_parameter_proc)(
    void* data,
    const char* name,
    const char* value);
```

| 表 21. コールバック・プロシージャのパラメーター |  |
|----------------------------|--|
| パラメーター                     | 説明   |
| data                       | <i>clemrtl_enumerateParameters()</i> に渡されるユーザー指定データ。 |
| name                       | パラメーター名。   |
| value                      | パラメーター値。   |

手順は、各イメージパラメーターに任意の順序で正確に一度適用されます。

### getParameter

```
int clemrtl_getParameter(
    clemrtl_image_handle handle,
    const char* name,
    char* value,
    int value_size);
```

イメージパラメーターの値を取得します。

| 表 22. <i>getParameter</i> パラメーター |                    |
|----------------------------------|--------------------|
| パラメーター                           | 説明                 |
| handle                           | イメージ ハンドル。         |
| name                             | パラメーター名。           |
| value                            | パラメーター値を受け取るバッファー。 |
| value_size                       | 値バッファーのサイズ。        |

次の表にあるステータス・コードのいずれかを返します。



| 表 23. ステータス・コード |                  |
|-----------------|------------------|
| 結果              | 説明               |
| CLEMRTL_OK      | 成功。              |
| CLEMRTL_FAIL    | エラー（追加情報はありません）。 |
| CLEMRTL_ERROR   | エラー（追加情報があります）。  |

パラメーター名がイメージ内のどのパラメーターの名前とも一致しない場合、または、終結のヌル文字を含むパラメーター値を受け取るに足りるほど値バッファが大きい場合、操作は失敗します。パラメーター名は、上記の `name.attribute` という形式 (`file0.name` など) を使用し、パラメーター値は、変換処理が数値の場合でも、常に文字列です。

### setParameter

```
int clemrtl_setParameter(
    clemrtl_image_handle handle,
    const char* name,
    const char* value);
```

イメージパラメーターの値を設定します。

| 表 24. setParameter パラメーター |           |
|---------------------------|-----------|
| パラメーター                    | 説明        |
| handle                    | イメージハンドル。 |
| name                      | パラメーター名。  |
| value                     | パラメーター値。  |

次の表にあるステータス・コードのいずれかを返します。

| 表 25. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

イメージハンドルが無効であったり、パラメーター名がイメージ内のどのパラメーターの名前とも一致しなかったりする場合、操作は失敗します。パラメーター名は、上記の `name.attribute` という形式 (`file0.name` など) を使用し、パラメーター値は、変換処理が数値の場合でも、常に文字列です。

### getFieldCount

```
int clemrtl_getFieldCount(
    clemrtl_image_handle handle,
    const char* key,
    size_t* field_count);
```

入力ソースまたは出力対象のフィールド数を返します。

| 表 26. getFieldCount パラメーター |           |
|----------------------------|-----------|
| パラメーター                     | 説明        |
| handle                     | イメージハンドル。 |

| 表 26. getFieldCount パラメーター (続き) |   |
|---------------------------------|---|
| パラメーター                          | 説明  |
| key                             | パラメーター・ファイルで使用されているとおりの、検証される入力または出力名。key はファイルまたはデータベースを参照する場合があります。 |
| field_count                     | フィールド数を取得します。   |

次の表にあるステータス・コードのいずれかを返します。

| 表 27. ステータス・コード |                   |
|-----------------|-------------------|
| 結果              | 説明                |
| CLEMRTL_OK      | 成功。               |
| CLEMRTL_FAIL    | エラー (追加情報はありません)。 |
| CLEMRTL_ERROR   | エラー (追加情報があります)。  |

### getFieldTypes

```
int clemrtl_getFieldTypes(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int* field_types);
```

入力ソースまたは出力対象のフィールド・タイプを返します。

| 表 28. getFieldTypes パラメーター |   |
|----------------------------|---|
| パラメーター                     | 説明  |
| handle                     | イメージハンドル。   |
| key                        | パラメーター・ファイルで使用されているとおりの、検証される入力または出力名。key はファイルまたはデータベースを参照する場合があります。   |
| field_count                | 検証するフィールド数。field_types 配列の長さ以下である必要があります。   |
| field_types                | フィールド・タイプを受け取る field_count 以上の配列の長さ。入力または出力の最初の field_count のタイプは配列にコピーされます。field_count が実際のフィールド数を超える場合、配列内の追加の要素は未定義のままです。フィールド・タイプの値は、次のデータ型テーブルに表示されています。 |

### データの型

field\_types は、次の表に表示されているデータ型のいずれかにする必要があります。

| 表 29. データ・タイプ |                         |             |
|---------------|-------------------------|-------------|
| データ型          | 解釈                      | 「C」での標準の宣言子 |
| STRING        | UTF-8 の、終わりにヌル文字を含む文字列。 | const char* |
| INTEGER       | 32 ビットの符号の付いた整数。        | int         |

表 29. データ・タイプ (続き)

| データ型      | 解釈   | 「C」での標準の宣言子 |
|-----------|--|-------------|
| LONG      | 64 ビットの符号の付いた整数。                           | long long   |
| REAL      | 64 ビットの浮動小数点。                              | double      |
| TIME      | 64 ビットの符号付き整数 (0 時以降の秒数)。                  | long long   |
| DATE      | 64 ビットの符号付き整数 (1970 年 1 月 1 以降の 0 時以降の秒数)。 | long long   |
| TIMESTAMP | 64 ビットの符号付き整数 (1970 年 1 月 1 以降の 0 時以降の秒数)。 | long long   |

次の表にあるステータス・コードのいずれかを返します。

表 30. ステータス・コード

| 結果            | 説明                |
|---------------|-------------------|
| CLEMRTL_OK    | 成功。               |
| CLEMRTL_FAIL  | エラー (追加情報はありません)。 |
| CLEMRTL_ERROR | エラー (追加情報があります)。  |

### setAlternativeInput

```
int clemrtl_setAlternativeInput(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int field_types,
    void** (*iterator)(void* arg),
    void* arg);
```

ファイル入力ソースを代替入力ソースと置き換えます。

表 31. setAlternativeInput パラメーター

| パラメーター      | 説明  |
|-------------|---|
| handle      | イメージハンドル。   |
| key         | パラメーター・ファイルで使用された元の入力ソースの名前。入力ソースはファイルである必要があるため、key は常に整数 $N \geq 0$ の「fileN」のような形式になります。  |
| field_count | 入力のフィールド数。値は、元のファイル入力のフィールド数と正確に一致する必要があります。一致しない場合はコールが失敗します。  |
| field_types | field_count 以上の長さの配列のタイプ。値 field_types[i] は、i 番目のフィールドのタイプで、以下のデータ型表に表示された値のいずれかである必要があります。タイプは、元のファイル入力のタイプに対応する必要があります。一致しない場合はコールが失敗します。 |

表 31. setAlternativeInput パラメーター (続き)

| パラメーター   | 説明  |
|----------|---|
| iterator | <p>代替入力データを作成する関数。関数は、次のように引数に適用されます。</p> <pre>void** row = iterator(arg);</pre> <p>関数は実行時に呼び出され (clemrtl_execute への呼び出し内)、入力レコードごとに 1 回呼び出されます。NULL の戻り値は、関数が再度呼び出されず実行が終了する場合の入力の終わりを示します。それ以外の場合は、field_count 以上の長さのデータの配列となり、row[i] では i 番目のフィールドの値を提供します。値は NULL の場合があります。そうでない場合は、タイプが対応する field_types[i] によって決定するデータへのポインタである必要があります。ポインタは、反復子の次の呼び出し、または実行が早めに終了する場合は実行の終了まで有効である必要があります。</p> |
| arg      | 各呼び出しの反復子に渡される明確でない引数。  |

### データの型

field\_types は、制限されたリスト内のいずれかのデータ型である必要があります。詳しくは、[28 ページの『getFieldTypes』](#)のトピックを参照してください。

次の表にあるステータス・コードのいずれかを返します。

| 表 32. ステータス・コード |                   |
|-----------------|-------------------|
| 結果              | 説明                |
| CLEMRTL_OK      | 成功。               |
| CLEMRTL_FAIL    | エラー (追加情報はありません)。 |
| CLEMRTL_ERROR   | エラー (追加情報があります)。  |

フィールド・タイプが元のファイル入力のフィールドタイプに対応していない場合、またはフィールド カウントが元のファイル入力のフィールド数と正確に一致しない場合、操作は失敗します。

### setAlternativeOutput

```
int clemrtl_setAlternativeOutput(
    clemrtl_image_handle handle,
    const char* key,
    size_t field_count,
    int field_types,
    void (*iterator)(void* arg, void** row),
    void* arg);
```

出力対象を代替出力対象と置き換えます。

| 表 33. setAlternativeOutput パラメーター |  |
|-----------------------------------|--|
| パラメーター                            | 説明   |
| handle                            | イメージ ハンドル。   |
| key                               | パラメーター・ファイルで使用された元の出力対象の名前。出力対象はファイルである必要があるため、key は常に整数 $N \geq 0$ の「fileN」のような形式になります。 |

表 33. setAlternativeOutput パラメーター (続き)

| パラメーター      | 説明   |
|-------------|--|
| field_count | 出力のフィールド数。値は、元のファイル出力のフィールド数と正確に一致する必要があります。一致しない場合は呼び出しが失敗します。  |
| field_types | field_count 以上の長さの配列のタイプ。値 field_types[i] は、i 番目のフィールドのタイプで、以下のデータ型表に表示された値のいずれかである必要があります。タイプは、元のファイル出力のタイプに対応する必要があります。一致しない場合はコールが失敗します。  |
| iterator    | <p>イメージ出力を使用する関数。この関数は、次のように引数およびデータ行に適用されます。</p> <pre>iterator(arg, row);</pre> <p>関数は実行時に呼び出され (clemrtl_execute への呼び出し内)、イメージに作成された結果行ごとに 1 回呼び出されます。NULL の row 値は、関数が再度呼び出されない出力の終わりを示します。実行が終了すると、アプリケーションはこの最後の呼び出しに依存せず、外部リソースを取り消して閉じます。それ以外の場合は、row は field_count 以上の長さのデータの配列となり、row[i] では i 番目の結果の値を提供します。値は NULL の場合があります、そうでない場合は、タイプが対応する field_types[i] によって決定するデータへのポインタである必要があります。呼び出しが戻された後はメモリーを保存することができないため、関数は必要なデータ値をコピーする必要があります。</p> |
| arg         | 各呼び出しの反復子に渡される明確でない引数。   |

### データの型

field\_types は、制限されたリスト内のいずれかのデータ型である必要があります。詳しくは、[28 ページの『getFieldTypes』](#)のトピックを参照してください。

次の表にあるステータス・コードのいずれかを返します。

| 結果            | 説明                |
|---------------|-------------------|
| CLEMRTL_OK    | 成功。               |
| CLEMRTL_FAIL  | エラー (追加情報はありません)。 |
| CLEMRTL_ERROR | エラー (追加情報があります)。  |

フィールド・タイプが元のファイル出力のフィールドタイプに対応していない場合、またはフィールド カウントが元のファイル出力のフィールド数と正確に一致しない場合、操作は失敗します。

### execute

```
int clemrtl_execute(clemrtl_image_handle handle);
```

イメージを実行します。

| 表 35. <i>execute</i> パラメーター |            |
|-----------------------------|------------|
| パラメーター                      | 説明         |
| handle                      | イメージ ハンドル。 |

次の表にあるステータス・コードのいずれかを返します。

| 表 36. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

イメージが準備されていない場合、イメージは初めに現在のパラメーター値を使用して準備されます。イメージハンドルが無効であったり、準備中または実行中にエラーが発生したりした場合、操作は失敗します。実行操作が完了するまでコールは返されません。

### prepare

```
int clemrtl_prepare(clemrtl_image_handle handle);
```

実行のためのイメージを準備します。

| 表 37. <i>prepare</i> パラメーター |            |
|-----------------------------|------------|
| パラメーター                      | 説明         |
| handle                      | イメージ ハンドル。 |

次の表にあるステータス・コードのいずれかを返します。

| 表 38. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

イメージ ハンドルが無効であったり、準備中にエラーが発生したりした場合、操作は失敗します。

イメージは、実行する前に準備する必要があります。イメージの準備によって、パラメーター値がイメージにフリーズされます。まだイメージが準備されていない場合、`clemrtl_execute()` は、呼び出しごとに自動的にイメージを準備します。イメージを同じパラメーター値で複数回実行する場合、`clemrtl_prepare()` を使用してイメージを準備するのが便利で、大幅にパフォーマンスを向上させる可能性があります。イメージが準備されると、パラメーター値への後続の変更は無視されます。`clemrtl_prepare()` を再度使用して新しいパラメーター値でイメージを更新します。

### interrupt

```
int clemrtl_interrupt(clemrtl_image_handle handle);
```

進行中のイメージの実行を終了します。

| 表 39. <i>interrupt</i> パラメーター |            |
|-------------------------------|------------|
| パラメーター                        | 説明         |
| handle                        | イメージ ハンドル。 |

次の表にあるステータス・コードのいずれかを返します。

| 表 40. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

イメージハンドルが無効であったり、イメージが実行しなかったりした場合、操作は効果を発揮しません。この関数は、同じイメージハンドル上でもう 1 つの API コールで同時にコールしても問題ありません。

### getErrorDetail

```
int clemrtl_getErrorDetail(
    clemrtl_image_handle handle,
    char* severity,
    int* code,
    char* text,
    int text_size);
```

イメージ上で発生した最後のエラーについて、詳細な情報を取得します。

| 表 41. getErrorDetail パラメーター |   |
|-----------------------------|---|
| パラメーター                      | 説明  |
| handle                      | イメージハンドル。   |
| severity                    | 重大度コードを単一文字で取得します。<br>I – 情報<br>W – 警告<br>E – エラー<br>X – システム・エラー |
| code                        | エラー番号を受け取ります。   |
| text                        | メッセージ テキストを受け取るバッファー。   |
| text_size                   | テキスト・バッファーのサイズ。   |

次の表にあるステータス・コードのいずれかを返します。

| 表 42. ステータス・コード |                 |
|-----------------|-----------------|
| 結果              | 説明              |
| CLEMRTL_OK      | 成功。             |
| CLEMRTL_ERROR   | エラー（追加情報があります）。 |

イメージハンドルが無効の場合、結果は読めません。必要に応じてメッセージ テキストを切り詰め、テキスト・バッファーに収まるようにします。

API コールが CLEMRTL\_ERROR で失敗し、エラーが特定のイメージハンドルに関係がない場合、0 をイメージハンドルとして通せば最後の非イメージ特定エラーの詳細を取得できますが、マルチスレッド環境ではこの方法は信頼できるものではありません。

## setReportHandler

```
int clemrtl_setReportHandler(  
    clemrtl_image_handle handle,  
    clemrtl_report_proc proc,  
    void* data);
```

イメージにレポート・コールバックの手順をインストールします。

| パラメーター | 説明                  |
|--------|---------------------|
| handle | イメージ ハンドル。          |
| proc   | レポート・コールバックの手順。     |
| data   | コールバックのためのユーザー指定データ |

次の表にあるステータス・コードのいずれかを返します。

| 結果            | 説明              |
|---------------|-----------------|
| CLEMRTL_OK    | 成功。             |
| CLEMRTL_ERROR | エラー（追加情報があります）。 |

コールバックの手順には次のような種類があります。

```
typedef void (*clemrtl_report_proc)(  
    void* data,  
    char severity,  
    int code,  
    const char* text);
```

| パラメーター   | 説明  |
|----------|---|
| data     | clemrtl_setReportHandler() に渡されるユーザー指定データ。                    |
| severity | 単一文字表示の重大度コード。<br>I – 情報<br>W – 警告<br>E – エラー<br>X – システム・エラー |
| code     | メッセージ番号。  |
| text     | メッセージ テキスト。   |

各メッセージの到達時に、手続きがメッセージに適用されます。NULL に手順を設定すると、イメージにインストールされた既存のハンドラが削除されます。



## 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にお願いします。

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。

## [ア行]

アプリケーションの例 [3](#)  
アンインストール  
    IBM SPSS Modeler Solution Publisher Runtime [18](#)  
一時ディレクトリー  
    IBM SPSS Modeler Solution Publisher Runtime [18](#)  
エラー・コード [21](#)  
エラーの報告  
    API 関数のリファレンス [33, 34](#)

## [サ行]

時間区域  
    API を使用した設定 [22](#)  
実行時  
    アンインストール [18](#)  
    一時ディレクトリー [18](#)  
    オプション [9](#)  
    開始 [17](#)  
    UNIX インストール [15](#)  
    Windows でのインストール [15](#)  
実例  
    アプリケーション ガイド [3](#)  
    概要 [4](#)  
出力ノード [5, 19](#)  
初期化  
    API プロセスの概要 [21](#)  
ステータス・コード [21](#)  
ストリームの公開  
    IBM SPSS Modeler Solution Publisher [5, 6](#)  
ストリームの実行  
    IBM SPSS Modeler Solution Publisher を使用した [9, 10](#)

## [タ行]

展開 [5, 19](#)  
閉じる  
    ストリーム [25](#)

## [ナ行]

ノードの公開 [19](#)

## [ハ行]

パラメーター  
    ストリーム実行用のパラメーター [10](#)  
    API イメージ パラメーター [25-27](#)  
    API コールバックの手順 [25](#)  
    API パラメーター・ファイル [24, 25](#)  
開く  
    ストリーム [24](#)

文書 [3](#)

## [ラ行]

ランタイム・プログラミング・ライブラリー (CLEMRTL) [12](#)  
ロケール  
    API を使用した設定 [22](#)

## A

API  
    エラー・コード [21](#)  
    概要 [12, 21](#)  
    実行オプション [23, 24](#)  
    ステータス・コード [21](#)  
    代替出力 [30](#)  
    代替入力 [29](#)  
    フィールド カウント [27](#)  
    フィールド・タイプ [28](#)  
    リファレンス [21](#)  
    setReportHandler [34](#)  
API 関数  
    初期化 [21, 22](#)  
    closeImage [21, 25](#)  
    enumerateParameters [21, 25](#)  
    execute [21, 31](#)  
    getErrorDetail [21, 33](#)  
    getFieldCount [21, 27](#)  
    getFieldTypes [21, 28](#)  
    getOption [21, 23](#)  
    getParameter [21, 26](#)  
    interrupt [21, 32](#)  
    openImage [21, 24](#)  
    prepare [21, 32](#)  
    setAlternativeInput [21, 29](#)  
    setAlternativeOutput [21, 30](#)  
    setLogFile [21](#)  
    setOption [21, 24](#)  
    setParameter [21, 27](#)  
    setReportHandler [21](#)

## C

closeImage  
    API 関数のリファレンス [25](#)  
    API プロセスの概要 [21](#)

## E

enumerateParameters  
    API 関数のリファレンス [25](#)  
    API プロセスの概要 [21](#)  
EPM  
    API [19](#)  
execute  
    API 関数のリファレンス [31](#)

execute (続き)  
API プロセスの概要 [21](#)

## G

getErrorDetail  
API 関数のリファレンス [33](#)  
API プロセスの概要 [21](#)

getFieldCount  
API 関数のリファレンス [27](#)  
API プロセスの概要 [21](#)

getFieldTypes  
API 関数のリファレンス [28](#)  
API プロセスの概要 [21](#)

getOption  
API 関数のリファレンス [23](#)  
API プロセスの概要 [21](#)

getParameter  
API 関数のリファレンス [26](#)  
API プロセスの概要 [21](#)

## I

IBM SPSS Modeler  
文書 [3](#)

IBM SPSS Modeler Server [1](#)

IBM SPSS Modeler Solution Publisher  
アプリケーションへの埋め込み [12](#)  
概要 [5](#)

IBM SPSS Modeler Solution Publisher Runtime  
アンインストール [18](#)  
一時ディレクトリー [18](#)  
インストール [15](#)  
開始 [17](#)  
システム要件 [15](#)

IBM SPSS Modeler Solution Publisher ノード [19](#)

IBM SPSS 組み込み可能予測分析  
API [19](#)

initialise  
API 関数のリファレンス [22](#)  
flags [22](#)

initialise\_ext  
API 関数のリファレンス [22](#)  
flags [22](#)

interrupt  
API 関数のリファレンス [32](#)  
API プロセスの概要 [21](#)

## O

openImage  
API 関数のリファレンス [24](#)  
API プロセスの概要 [21](#)

## P

prepare  
API 関数のリファレンス [32](#)  
API プロセスの概要 [21](#)

## S

setAlternativeInput

setAlternativeInput (続き)  
API 関数のリファレンス [29](#)  
API プロセスの概要 [21](#)

setAlternativeOutput  
API 関数のリファレンス [30](#)  
API プロセスの概要 [21](#)

setLogFile  
API プロセスの概要 [21](#)

setOption  
API 関数のリファレンス [24](#)  
API プロセスの概要 [21](#)

setParameter  
API 関数のリファレンス [27](#)  
API プロセスの概要 [21](#)

setReportHandler  
API 関数のリファレンス [34](#)  
API プロセスの概要 [21](#)

## U

UNIX

IBM SPSS Modeler Solution Publisher Runtime のインストール [15](#)

## [特殊文字]

.par ファイル [6](#)

.pim ファイル [6](#)



