

*IBM SPSS Modeler 18.6 Solution
Publisher*



Nota

Antes de utilizar essas informações e o produto que elas suportam, leia as informações em “[Avisos](#)” na [página 39](#).

Informações do produto

Esta edição se aplica à versão 18, release 4, modificação 0 de IBM® SPSS Modeler e a todos os lançamentos e modificações subsequentes até indicado de outra forma em novas edições.

© Copyright International Business Machines Corporation .

Índice

Prefácio.....	v
Capítulo 1. SobreIBM SPSS Modeler.....	1
Produtos IBM SPSS Modeler.....	1
IBM SPSS Modeler.....	1
ServidorIBM SPSS Modeler.....	1
Console de administraçãoIBM SPSS Modeler.....	2
IBM SPSS Modeler Batch.....	2
Editor de soluçõesIBM SPSS Modeler.....	2
ServidorIBM SPSS Modeler Adaptadores para Serviços de Colaboração e ImplementaçãoIBM SPSS.....	2
Edições do IBM SPSS Modeler.....	2
Documentação.....	3
Documentação do SPSS Modeler Professional.....	3
SPSS Modeler Premium documentação.....	4
Exemplos de Aplicação.....	4
Pasta Demos.....	4
Rastreamento de Licença.....	4
Capítulo 2. Editor de soluçõesIBM SPSS Modeler.....	5
Como o Editor de soluçõesIBM SPSS Modeler funciona.....	5
fluxos de publicação.....	6
Capítulo 3. Executando fluxos publicados.....	9
Usando o Arquivo de Parâmetro para Customizar a Execução de Fluxo.....	10
Incorporando o Editor de soluçõesIBM SPSS Modeler nos Aplicativos.....	12
Aplicativo Editor de soluçõesIBM SPSS Modeler Sample.....	13
Capítulo 4. Instalando Tempo de execuçãoIBM SPSS Modeler Solution Publisher..	15
Instalando o Tempo de Execução.....	15
Instalando no Windows.....	15
Instalando no UNIX.....	15
Resolvendo Problemas em uma Instalação.....	17
Iniciando o Tempo de Execução.....	17
Alterando o diretório temporário.....	18
Desinstalando Editor de soluçõesIBM SPSS Modeler Runtime.....	18
Procedimento de Desinstalação do Windows.....	18
Procedimento de Desinstalação do UNIX.....	19
Capítulo 5. IBM SPSS Embedded Predictive Modeling API.....	21
Capítulo 6. Editor de soluçõesIBM SPSS Modeler Referência da API da Biblioteca de Tempo de Execução.....	23
Introdução.....	23
Visão Geral do Processo da API.....	23
Funções de API.....	24
Avisos.....	39
Marcas comerciais.....	40

Termos e condições para documentação do produto.....	40
Índice remissivo.....	43

Prefácio

IBM SPSS Modeler é o ambiente de trabalho de mineração de dados de força corporativa do IBM Corp. . O ModeladorSPSS ajuda as organizações a melhorarem as relações com o cliente e com o cidadão por meio de um entendimento profundo dos dados. As organizações utilizam o insight adquirido do ModeladorSPSS para reter clientes rentáveis, identificar oportunidades de venda cruzada, atrair novos clientes, detectar fraude, reduzir o risco e melhorar a entrega de serviço de governo.

A interface visual do ModeladorSPSS convida os usuários a aplicarem seus conhecimentos de negócios específicos, levando a modelos preditivos mais poderosos e reduzindo o tempo para a solução. O ModeladorSPSS oferece muitas técnicas de modelagem, como predição, classificação, segmentação e algoritmos de detecção de associação. Quando os modelos são criados, o Editor de soluçõesIBM SPSS Modeler permite entregá-los aos tomadores de decisão na empresa ou a um banco de dados.

Sobre o IBM Business Analytics

O software IBM Business Analytics fornece informações completas, consistentes e exatas nas quais os tomadores de decisão confiam para melhorar o desempenho de negócios. Um portfólio abrangente de inteligência de negócios, análise preditiva, gerenciamento de desempenho financeiro e estratégias e aplicativos analíticos fornecem insight claro, imediato e prático sobre o desempenho atual e a capacidade de prever resultados futuros. Combinado com soluções para segmentos do mercado, práticas comprovadas e serviços profissionais completos, organizações de qualquer tamanho poderão conduzir maior produtividade, automatizar as decisões de modo confiável e entregar melhores resultados.

Como parte deste dossier, o software IBM SPSS Predictive Analytics ajuda as organizações a prever futuros eventos e agir proativamente com esse insight para melhores resultados de negócios. Os clientes acadêmicos, comerciais e do governo no mundo todo se baseiam na tecnologia do IBM SPSS como uma vantagem competitiva para atrair, manter e aumentar seus clientes, enquanto reduz fraudes e minimiza riscos. Ao incorporar o software IBM SPSS em suas operações diárias, as organizações se tornam empresas preditivas – capazes de dirigir e automatizar decisões para atender às metas de negócio e obter vantagem competitiva mensurável. Para obter mais informações ou entrar em contato com um representante, visite <https://www.ibm.com/mysupport/s/>.

Supporte técnico

O suporte técnico está disponível para manutenção dos clientes. Os clientes podem entrar em contato com o Suporte Técnico para obter assistência no uso de produtos IBM Corp. ou para obter ajuda na instalação em um dos ambientes de hardware suportados. Para obter o Suporte Técnico, consulte o website do IBM Corp. em <https://www.ibm.com/mysupport/s/>. Esteja preparado para se identificar, sua organização e seu contrato de suporte ao solicitar assistência.

Capítulo 1. SobreIBM SPSS Modeler

O IBM SPSS Modeler é um conjunto de ferramentas de mineração de dados que permite desenvolver rapidamente modelos preditivos usando o conhecimento de negócios, e implementá-los em operações de negócios para melhorar a tomada de decisão. Projetado em torno do modelo CRISP-DM padrão de mercado, o IBM SPSS Modeler suporta todo o processo de mineração de dados, a partir dos dados para melhores resultados de negócios.

OIBM SPSS Modeler oferece uma variedade de métodos de modelagem tomados do aprendizado de máquina, inteligência artificial e estatística. Os métodos disponíveis na paleta Modelagem permitem derivar informações novas a partir dos dados, e desenvolver modelos preditivos. Cada método possui certas forças e é mais adequado para certos tipos de problemas.

O ModeladorSPSS pode ser comprado como um produto independente, ou usado como um cliente na combinação com o Servidor doSPSS Modeler. Várias opções adicionais também estão disponíveis, conforme resumidas nas seções a seguir. Para mais informações, consulte <https://www.ibm.com/analytics/us/en/technology/spss/>.

Produtos IBM SPSS Modeler

A família de produtos IBM SPSS Modeler e o software associado abrangem o seguinte.

- IBM SPSS Modeler
- ServidorIBM SPSS Modeler
- Console de administraçãoIBM SPSS Modeler (incluído com IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- Editor de soluçõesIBM SPSS Modeler
- ServidorIBM SPSS Modeler adaptadores para Serviços de Colaboração e ImplementaçãoIBM SPSS

IBM SPSS Modeler

ModeladorSPSS é uma versão funcionalmente completa do produto que você instala e executa em seu computador pessoal. É possível executar o ModeladorSPSS no modo local como um produto independente ou usá-lo no modo distribuído com ServidorIBM SPSS Modeler para melhorar o desempenho em conjuntos de dados grandes.

Com o ModeladorSPSS, é possível construir modelos preditivos exatos de maneira rápida e intuitiva, sem programação. Usando a interface visual exclusiva, é possível visualizar facilmente o processo de mineração de dados. Com o suporte da análise avançada integrada ao produto, é possível descobrir tendências e padrões ocultos anteriormente em seus dados. É possível modelar resultados e entender os fatores que os influenciam, permitindo que você aproveite as vantagens das oportunidades de negócios e diminua os riscos.

ModeladorSPSS está disponível em duas edições: SPSS Modeler Professional e SPSS Modeler Premium. Consulte o tópico “[Edições do IBM SPSS Modeler](#)” na página 2 para obter mais informações.

ServidorIBM SPSS Modeler

ModeladorSPSS usa uma arquitetura de cliente/servidor para distribuir solicitações para operações cheias de recursos para poderosos softwares de servidor, resultando em desempenho mais rápido em conjuntos de dados maiores.

Servidor doSPSS Modeler é um produto licenciado separadamente que é executado de forma contínua no modo de análise distribuído em um host do servidor com uma ou mais instalações do IBM SPSS Modeler. Dessa maneira, o Servidor doSPSS Modeler fornece desempenho superior em conjuntos de dados grandes, pois operações com uso intensivo de memória podem ser executadas no servidor sem

fazer download dos dados no computador cliente. ServidorIBM SPSS Modeler também fornece suporte para otimização de SQL e capacidades de modelagem dentro da base de dados, entregando mais benefícios para o desempenho e a automação.

Console de administraçãoIBM SPSS Modeler

O Console de administração do Modeler é uma interface gráfica de usuário para o gerenciamento de muitas das opções de configuração Servidor doSPSS Modeler , que também são configuráveis por meio de um arquivo de opções. O console é incluído em IBM SPSS Deployment Manager, pode ser usado para monitorar e configurar suas instalações Servidor doSPSS Modeler , e está disponível gratuitamente para os clientes atuais Servidor doSPSS Modeler . O aplicativo pode ser instalado somente em computadores Windows; no entanto, ele pode administrar um servidor instalado em qualquer plataforma suportada.

IBM SPSS Modeler Batch

Embora geralmente a mineração de dados seja um processo interativo, também é possível executar o ModeladorSPSS a partir de uma linha de comandos, sem a necessidade de uma interface gráfica com o usuário. Por exemplo, você pode ter tarefas repetidas ou de longa execução que deseja executar sem intervenção do usuário. SPSS Modeler Batch é uma versão especial do produto que fornece suporte para capacidades de análise completa do ModeladorSPSS sem acessar a interface com o usuário regular. Servidor doSPSS Modeler é necessário para usar o SPSS Modeler Batch.

Editor de soluçõesIBM SPSS Modeler

Editor de soluçõesSPSS Modeler é uma ferramenta que permite criar uma versão do pacote de um fluxo do ModeladorSPSS que pode ser executado por um mecanismo de tempo de execução externo ou integrado a um aplicativo externo. Dessa maneira, é possível publicar e implementar fluxos completos do ModeladorSPSS para uso em ambientes que não têm o ModeladorSPSS instalado. Editor de soluçõesSPSS Modeler é distribuído como parte do serviço IBM SPSS Collaboration and Deployment Services - Pontuação, para o qual uma licença separada é necessária. Com essa licença, você recebe o Tempo de execuçãoSPSS Modeler Solution Publisher, que permite executar os fluxos publicados.

Para obter mais informações sobre Editor de soluçõesSPSS Modeler, consulte a documentação do Serviços de Colaboração e ImplementaçãoIBM SPSS. A Serviços de Colaboração e ImplementaçãoIBM SPSS IBM Documentation contém seções chamadas "IBM SPSS Modeler Solution Publisher" e "IBM SPSS Analytics Toolkit."

ServidorIBM SPSS Modeler Adaptadores para Serviços de Colaboração e ImplementaçãoIBM SPSS

Inúmeros adaptadores para o Serviços de Colaboração e ImplementaçãoIBM SPSS estão disponíveis para permitir que o ModeladorSPSS e o Servidor doSPSS Modeler interajam com um repositório do Serviços de Colaboração e ImplementaçãoIBM SPSS. Dessa forma, um fluxo do ModeladorSPSS implementado no repositório pode ser compartilhado por diversos usuários ou acessado a partir do aplicativo thin client VantagensIBM SPSS Modeler. Você instala o adaptador no sistema que hospeda o repositório.

Edições do IBM SPSS Modeler

ModeladorSPSS está disponível nas seguintes edições.

SPSS Modeler Professional

SPSS Modeler Professional fornece todas as ferramentas necessárias para você trabalhar com a maioria dos tipos de dados estruturados, como comportamentos e interações controlados em sistemas CRM, demográficos, comportamento de compra e dados de vendas.

SPSS Modeler Premium

SPSS Modeler Premium é um produto licenciado separadamente que se estende SPSS Modeler Professional para trabalhar com dados especializados e com dados de texto não estruturados. SPSS Modeler Premium inclui Análise de texto doIBM SPSS Modeler:

Análise de texto doIBM SPSS Modeler usa tecnologias de linguística avançada e processamento de linguagem natural (NLP) para processar rapidamente uma grande variedade de dados de texto não estruturados, extrair e organizar conceitos chave e agrupar esses conceitos em categorias. Categorias e conceitos extraídos podem ser combinados com dados estruturados existentes, como demográficos, e aplicados à modelagem usando o conjunto completo de ferramentas de mineração de dados do IBM SPSS Modeler para gerar decisões melhores e mais focadas.

AssinaturaIBM SPSS Modeler

AssinaturaIBM SPSS Modeler fornece todas as mesmas capacidades de analítica preditiva que o cliente IBM SPSS Modeler tradicional. Com a edição de Assinaturas, é possível fazer o download de atualizações do produto regularmente.

Documentação

A documentação está disponível no menu **Ajuda** em ' ModeladorSPSS. Isso abre a IBM Documentation on-line, que está sempre disponível fora do produto.

A documentação completa de cada produto (incluindo instruções de instalação) também está disponível em formato PDF. Consulte a página de suporte a seguir: [**DocumentaçãoSPSS Modeler 18.6**](#).

Documentação do SPSS Modeler Professional

O conjunto de documentações do SPSS Modeler Professional (excluindo instruções de instalação) é o seguinte.

- **IBM SPSS Modeler User's Guide.** Introdução geral para usar ModeladorSPSS, incluindo como construir fluxos de dados, manipular valores ausentes, construir expressões CLEM , trabalhar com projetos e relatórios, e streams de pacotes para implementação em Serviços de Colaboração e ImplementaçãoIBM SPSS ou VantagensIBM SPSS Modeler.
- **Nós de Origem, de Processo e de Saída do IBM SPSS Modeler.** Descrições de todos os nós usados para ler, processar e emitir dados em diferentes formatos. Efetivamente, isso significa todos os nós além dos de modelagem.
- **Nós de Modelagem do IBM SPSS Modeler.** Descrições de todos os nós usados para criar modelos de mineração de dados. OIBM SPSS Modeler oferece uma variedade de métodos de modelagem tomados do aprendizado de máquina, inteligência artificial e estatística.
- **Guia de Aplicativos do IBM SPSS Modeler.** Os exemplos neste guia fornecem introduções sintetizadas e direcionadas para técnicas e métodos de modelagem específicos. Uma versão online deste guia também está disponível no menu Ajuda. Veja o tópico “[Exemplos de Aplicação](#)” na página 4 para obter mais informações.
- **Script e Automação Python do IBM SPSS Modeler.** Informações sobre como automatizar o sistema por meio de script Python, incluindo as propriedades que podem ser usadas para manipular nós e fluxos.
- **Guia de Implementação do IBM SPSS Modeler.** Informações sobre a execução de fluxos IBM SPSS Modeler como etapas de processamento de tarefas sob IBM SPSS Deployment Manager.
- **Guia de Mineração Dentro do Banco de Dados do IBM SPSS Modeler.** Informações sobre como usar o poder do seu banco de dados para melhorar o desempenho e ampliar o intervalo de capacidades analíticas por meio de algoritmos de terceiros.
- **Guia de Desempenho e de Administração do ServidorIBM SPSS Modeler.** Informações sobre como configurar e administrar o ServidorIBM SPSS Modeler.

- **Guia do Usuário do IBM SPSS Deployment Manager.** Informações sobre o uso da interface de usuário do console de administração incluídas no aplicativo Gerente de implantação para monitoramento e configuração ServidorIBM SPSS Modeler.
- **IBM SPSS Modeler Guia CRISP-DM.** Guia passo a passo para o uso da metodologia CRISP-DM para mineração de dados com ModeladorSPSS.
- **IBM SPSS Modeler Batch User's Guide.** Guia completo para o uso do IBM SPSS Modeler no modo em lote, incluindo detalhes da execução do modo em lote e argumentos de linha de comandos. Este guia está disponível somente em formato PDF.

SPSS Modeler Premium documentação

O conjunto de documentações do SPSS Modeler Premium (excluindo instruções de instalação) é o seguinte.

- **Análise de texto do SPSS Modeler User's Guide.** Informações sobre o uso de analítica de texto com ModeladorSPSS, cobrindo os nós de mineração de texto, ambiente de trabalho interativo, modelos e outros recursos.

Exemplos de Aplicação

Enquanto as ferramentas de mineração de dados no ModeladorSPSS podem ajudar a resolver uma ampla variedade de negócios e problemas organizacionais, os exemplos de aplicativos fornecem introduções breves e destinadas aos métodos e técnicas de modelagem específicos. Os conjuntos de dados utilizados aqui são muito menores do que as enormes lojas de dados gerenciadas por alguns mineiros de dados, mas os conceitos e métodos que estão envolvidos são escaláveis para aplicações do mundo real.

Para acessar os exemplos, clique em **Exemplos de aplicativos** no menu Ajuda em ModeladorSPSS.

Os arquivos de dados e os fluxos de amostra são instalados na pasta Demos no diretório de instalação do produto. Para obter mais informações, consulte “[Pasta Demos](#)” na página 4.

Exemplos de modelagem da base de dados. Consulte os exemplos no *Guia de Mineração dentro do Banco de Dados do IBM SPSS Modeler*.

Exemplos de script. Consulte os exemplos no *Guia de Script e Automação do IBM SPSS Modeler*.

Pasta Demos

Os arquivos de dados e fluxos de amostra que são utilizados com os exemplos de aplicação são instalados na pasta Demos sob o diretório de instalação do produto (por exemplo: C:\Program Files\IBM\SPSS\Modeler\<version>\Demos). Esta pasta também pode ser acessada a partir do grupo de programas IBM ModeladorSPSS no menu Iniciar do Windows, ou clicando em Demos na lista de diretórios recentes na caixa de diálogo **Arquivo > Open Stream**.

Rastreamento de Licença

Quando você usa o ModeladorSPSS, o uso sob licença é controlado e registrado em intervalos regulares. As métricas de licença que são registradas são *AUTHORIZED_USER* e *CONCURRENT_USER* e o tipo de métrica que é registrado depende do tipo de licença que você possui para o ModeladorSPSS.

Os arquivos de log que são produzidos podem ser processados pelo IBM License Metric Tool, do qual é possível gerar relatórios de uso sob licença.

Os arquivos de log de licença são criados no mesmo diretório onde os arquivos de log do Client log do ModeladorSPSS são registrados (por padrão, %ALLUSERSPROFILE%\IBM\SPSS\Modeler\<version>\log).

Capítulo 2. Editor de soluçõesIBM SPSS Modeler

Editor de soluçõesIBM SPSS Modeler é uma ferramenta poderosa para integrar seus resultados de mineração de dados em seu processo de negócios para solucionar os problemas do mundo real. Usando o Editor de soluçõesIBM SPSS Modeler, é possível criar uma versão "empacotada" de um fluxo que pode ser executado por um mecanismo de Tempo de Execução externo ou integrado no aplicativo externo. Isso permite que você implemente seus fluxos de modelagem de dados em um ambiente de produção para suportar seus processos de negócios diários e para conferir poderes aos tomadores de decisões da sua organização com o conhecimento obtido da mineração de dados.

Usar o Editor de soluçõesIBM SPSS Modeler oferece mais poder do que simplesmente exportar o modelo (como PMML), porque ele permite que você publique e implemente *fluxos de IBM SPSS Modeler completos*. Isso significa que você pode executar a preparação de dados, bem como as operações de campo e de registro, como agregando dados, selecionando registros ou derivando novos campos antes de criar previsões com base em um modelo. Em seguida, é possível processar os resultados do modelo antes de salvar os dados--tudo isso simplesmente executando o fluxo publicado.

Nota: A publicação é obtida usando a guia Publish do nó de exportação apropriado. Consulte o tópico “[fluxos de publicação](#)” na página 6 para obter informações adicionais. Fluxos salvos nas liberações anteriores a versão 11.0 do produto usaram um nó de Publicação separado. O nó ainda é suportado para compatibilidade com versões anteriores, mas não está mais disponível na paleta do nó.

LicenciamentoEditor de soluçõesIBM SPSS Modeler

Nota: Editor de soluçõesIBM SPSS Modeler é distribuído como parte do Serviços de Colaboração e ImplementaçãoIBM SPSS Serviço de Scoring, para o qual é necessária uma licença separada. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Ao comprar uma licença, um Tempo de execuçãoIBM SPSS Modeler Solution Publisher separado é fornecido, o que permite que você execute os fluxos publicados. Veja o tópico [“Instalando o Tempo de Execução”](#) na página 15 para obter mais informações.

Criação de log

O arquivo de log é controlado pelo arquivo de configuração `log4cxx.properties`, localizado na pasta `config` do seu diretório de instalação Editor de soluçõesIBM SPSS Modeler. Você pode modificar o arquivo de configuração `log4cxx.properties` para customizar a saída de registro.

Note que se usando o Linux/UNIX, para que o aplicativo encontre o arquivo de configuração, você deve definir uma variável de ambiente antes de executar o aplicativo (por exemplo, `export MODELERRUNTIME=MSP_INSTALLATION_DIRECTORY`).

Como o Editor de soluçõesIBM SPSS Modeler funciona

Implementar uma solução usando o Editor de soluçõesIBM SPSS Modeler envolve duas fases: a *publicação* de um fluxo e a *execução* de um fluxo.

Publicação. Conforme você trabalha através do processo de mineração de dados, eventualmente, você chegará a um modelo que forneça uma boa solução para seu problema de negócios. Nesse ponto, você estará pronto para executar esse modelo e aplicá-lo em seu processo de negócios. Ao publicar um fluxo, uma descrição detalhada do fluxo é gravada no disco (como um arquivo de *imagem* e um arquivo de *parâmetro*). Consulte o tópico [“fluxos de publicação”](#) na página 6 para obter mais informações.

Execução. Após ter publicado o fluxo, é possível recriar o processo implementado no fluxo executando o fluxo publicado. Isso é feito usando o Tempo de Execução do IBM SPSS Modeler independente (`modelerrun.exe`) ou desenvolvendo um aplicativo que utilize a Biblioteca de Tempo de Execução do IBM SPSS Modeler para executar o fluxo. Para executar fluxos fora do IBM SPSS Modeler (usando o Tempo de Execução ou um aplicativo customizado), primeiro, você deve instalar o Tempo de execuçãoIBM SPSS

Modeler Solution Publisher. Consulte o tópico “Instalando o Tempo de Execução” na página 15 para obter mais informações.

fluxos de publicação

Os fluxos de publicação são feitos diretamente a partir de IBM SPSS Modeler usando qualquer um dos nós de exportação padrão: Banco de Dados, Arquivo Flat, Estatísticas Exportação, Exportação de Extensão, Exportação de Dados de Coleta de Dados, Exportação de SAS, Excel e XML Export. O tipo de nó de exportação determina o formato dos resultados a serem gravados todas as vezes em que o fluxo publicado for executado usando o Tempo de execução IBM SPSS Modeler Solution Publisher ou aplicativo externo. Por exemplo, se desejar gravar seus resultados em um banco de dados todas as vezes em que o fluxo publicado for executado, use um nó da Base de Dados.

Para publicar um fluxo

1. Abra ou construa um fluxo no modo normal e anexe um nó de exportação no término.
2. Na guia Publish no nó de exportação, especifique um nome de raiz para os arquivos publicados (ou seja, o nome do arquivo para o qual serão anexadas as extensões .pim, .par e .xml).
3. Clique em **Publicar** para publicar o fluxo, ou selecione **Publicar o fluxo** para publicar automaticamente o fluxo cada vez que o nó for executado.

Nome Publicado. Especificar o nome da raiz para os arquivos de imagem e parâmetro publicados.

- O **arquivo de imagem** (*.pim) fornece todas as informações necessárias para que o Tempo de Execução execute o fluxo publicado exatamente como foi no momento da exportação. Se você estiver confiante de que não precisará alterar nenhuma configuração para o fluxo (tal como a origem de dados de entrada ou o arquivo de dados de saída), é possível implementar apenas o arquivo de imagem.
- O **arquivo de parâmetro** (*.par) contém informações configuráveis sobre as origens de dados, arquivos de saída e opções de execução. Se você desejar poder controlar a entrada ou a saída do fluxo sem publicar novamente o fluxo, você precisará tanto do arquivo de parâmetro quanto do arquivo de imagem.
- O **arquivo de metadados** (*.xml) descreve as entradas e saídas da imagem e seus modelos de dados. É projetado para ser usado por aplicativos que integram a biblioteca de tempo de execução e que precisam conhecer a estrutura dos dados de entrada e de saída.

Nota: Este arquivo só é produzido se você selecionar a opção **Metadados Publish**.

Publicar parâmetros. Se necessário, é possível incluir parâmetros de fluxo no arquivo *.par. É possível alterar esses valores de parâmetro de fluxo ao executar a imagem editando o arquivo *.par ou por meio da API de tempo de execução.

Esta opção possibilita o botão **Parâmetros**. A caixa de diálogo Parâmetros de Publicação é exibida ao clicar no botão.

Escolha os parâmetros que deseja incluir na imagem publicada, selecionando a opção relevante na coluna **Publish**.

Na execução de fluxo. Especifica se o fluxo é automaticamente publicado quando o nó for executado.

- **Dados de exportação.** Executa o nó de exportação no modo padrão, sem publicar o fluxo. (Basicamente, o nó executa em IBM SPSS Modeler da mesma forma que seria se Editor de soluções IBM SPSS Modeler não estivesse disponível.) Se você selecionar esta opção, o fluxo não será publicado, a menos que você faça isso explicitamente clicando em **Publicar** na caixa de diálogo do nó de exportação. Como alternativa, é possível publicar o fluxo atual usando a ferramenta Publicar na barra de ferramentas ou usando um script.
- **Publicar o fluxo.** Publica o fluxo para implementação usando o Editor de soluções IBM SPSS Modeler. Seleccione essa opção se desejar publicar automaticamente o fluxo todas as vezes em que ele for executado.

Nota:

- Se você planeja executar o fluxo publicado com dados novos ou atualizados, é importante observar que a ordem dos campos no arquivo de entrada deve ser a mesma da ordem dos campos no arquivo de entrada do nó de origem especificada no fluxo publicado.
- Ao publicar em aplicativos externos, considere filtrar campos extras ou renomeados em conformidade com os requisitos de entrada. Ambos podem ser realizados usando um nó de Filtro antes do nó de exportação.

Capítulo 3. Executando fluxos publicados

O principal efeito da execução de um fluxo publicado é sempre gerar um conjunto de dados. Os dados podem ser armazenados em um arquivo de disco ou gravados em um banco de dados. Os fluxos publicados não podem gerar gráficos, tabelas, modelos gerados ou outra saída sem dados.

Os fluxos publicados usando Editor de soluções IBM SPSS Modeler podem ser executados usando o programa Tempo de execução IBM SPSS Modeler Solution Publisher . O programa Runtime, modelerrun.exe, é iniciado a partir da linha de comando, com opções indicando o arquivo de imagem e o arquivo de parâmetro (opcional) para execução. O comando de Tempo de Execução é o seguinte:

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

onde as entradas de comando são mostradas na tabela a seguir.

Tabela 1. Entradas de comando de Tempo de Execução	
Entrada de Comando	Descrição
-nobanner	suprime a mensagem de texto de inicialização para o Tempo de Execução (opcional).
<parameter-file>	é o nome do arquivo do arquivo de parâmetro (opcional).
<options>	é uma lista separada por comma de option=value pares especificando opções de execução para o Runtime. As opções são detalhadas abaixo (opcional). Observação: Você não deve deixar nenhum espaço em branco após uma vírgula.
<image-file>	é o nome do arquivo do arquivo de imagem a ser executado (obrigatório).

As opções de execução disponíveis são mostradas na tabela a seguir.

Tabela 2. Opções de Tempo de Execução Válido		
Nome	Valor padrão	Descrição
max_file_size	-1	Tamanho máximo dos arquivos salvos. Um valor -1 indica nenhum limite.
max_sql_string_length	2.048	Comprimento máximo de uma sequência de caracteres importada do banco de dados com SQL. Valores de sequência maiores que este estão truncados à direita sem aviso. O intervalo válido está entre 1 e 65.535 caracteres.
memory_usage	100	Multiplicador para alocação de memória dinâmica. Ajuste esse valor para cima ou para baixo para regular o uso de memória total do servidor.
temp_directory	"	Especifica o diretório a ser usado para arquivos temporários.

Tabela 2. Opções de Tempo de Execução Válido (continuação)

Nome	Valor padrão	Descrição
request_passwords	Y	Controla se você será solicitado por uma senha do banco de dados ao executar um fluxo publicado que requer acesso ao banco de dados (S/N).

Nota: Ao publicar um fluxo que contém um nó de entrada do usuário, observe que ModeladorSPSS Solução Publisher requer um arquivo de entrada para o nó (enquanto que em ModeladorSPSS você define o campo e seus dados diretamente no nó). O nome do arquivo padrão é userInput.dat, e você pode alterá-lo no arquivo de parâmetro.

Nota:

- Ao executar um fluxo contendo um nó de Otimização do CPLEX em **ModeladorSPSS Solution Publisher**, por padrão a biblioteca do CPLEX de edição comunitária incorporada é usada. Ele tem uma limitação de 1000 variáveis e 1000 restrições. Se você instalar a edição completa do IBM ILOG CPLEX e desejar usar o mecanismo do CPLEX da edição completa em vez disso, que não tem essas limitações, conclua a etapa a seguir para sua plataforma:

- No Windows, adiciona o caminho da biblioteca OPL como um argumento de linha de comando para modelerrun.exe. Por exemplo:

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

Onde <CPLEX_path> é o diretório de instalação do CPLEX como C:\Program Files\IBM\ILOG\CPLEX_Studio127, e <Platform_dir> é o diretório específico da plataforma, como x64_win64.

- No Linux, edite modelerrun e adiciona o caminho da biblioteca OPL. Por exemplo:

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

Onde <CPLEX_path> é o diretório de instalação do CPLEX como /root/Libs_127_FullEdition/Linux_x86_64, e <Platform_dir> é o diretório específico da plataforma, como x86-64_linux.

- O CPLEX não é suportado no macOS. Você pode usar o nó (adicioná-lo ao seu fluxo, editar suas propriedades, etc), mas você não pode executá-lo.

Executando um fluxo local Apache Spark em Editor de soluçõesIBM SPSS Modeler

Ao executar um fluxo Spark local em Editor de soluçõesIBM SPSS Modeler, você deve especificar o seguinte no comando modelerrun (onde C:/Anaconda3/python.exe é o caminho para o seu executável Python). A Anaconda 3 é necessária (Anaconda 2 não é suportada).

```
-o eas_pyspark_python_path=C:/Anaconda3/python.exe
```

Usando o Arquivo de Parâmetro para Customizar a Execução de Fluxo

Quando um fluxo é publicado, o arquivo de imagem contém informações sobre o destino dos dados de saída e da origem de dados que foram selecionados no fluxo original. No entanto, às vezes, é conveniente executar um fluxo com relação aos dados de uma origem de dados diferente ou rotear novamente os dados de saída para um destino diferente. Isso pode ser feito editando o arquivo de parâmetro do fluxo.

Isso é particularmente importante quando você estiver executando o fluxo publicado em um computador diferente daquele em que foi criado. Será necessário atualizar as localizações dos arquivos de entrada e de saída para refletir a estrutura do computador de destino. Observe que os fluxos publicados não suportam os caminhos relativos, o que torna especialmente importante verificar os locais dos dados em

seu arquivo de parâmetro. (Se você especificar caminhos relativos nas opções do nós de Exportação ao publicar o fluxo, o IBM SPSS Modeler irá convertê-los em caminhos absolutos antes de criar os arquivos de parâmetro e de imagem para o fluxo publicado.)

Também é possível usar o arquivo de parâmetro para configurar as opções de execução e para alterar os valores dos parâmetros de fluxo.

Os valores de parâmetro estão especificados no arquivo de parâmetro como pares do <name.attribute>=<value>. Para alguns parâmetros, o nome do parâmetro contém um dígito de ID para distinguir os nós do mesmo tipo no fluxo. Por exemplo, em um fluxo em que tanto a entrada quanto a saída referem-se a uma conexão com o banco de dados, os parâmetros descrevendo a conexão de entrada podem ser dbconn0.datasource, dbconn0.user, etc., e os parâmetros associados à conexão de saída podem ser dbconn1.datasource, dbconn1.user, etc. Os valores são sempre fechados em citações duplas. O arquivo de parâmetro contém os seguintes parâmetros, todos os quais podem ser alterados para customizar a execução de fluxo.

Tabela 3. Parâmetros de execução	
Parâmetro	Propósito
dbconnx.datasource	Nome da origem de dados (DSN).
dbconnx.user	Nome do usuário para acesso restrito aos bancos de dados.
dbconnx.password	Senha para acesso restrito aos banco de dados.
dbconnx.epassword	Senha codificada para acesso restrito aos banco de dados. Para gerar uma senha codificada, selecione Codificar Senha no menu Ferramentas da interface com o usuário IBM SPSS Modeler. Copie e cole a senha codificada, conforme necessário.
dbconnx.password_required	Sinalização que indica se é necessário solicitar (prompt) uma senha para essa conexão com o banco de dados. (O valor é "Y" ou "N".) Configurado automaticamente para "N" quando não houver nenhuma senha especificada no momento da publicação.
dbconnx.catalog	Catálogo associado à conexão com o banco de dados.
dbtablex.name	Nome da tabela base. <i>Observação:</i> A ordem dos campos em uma nova tabela deve ser a mesma da ordem dos campos no fluxo originalmente publicado.
dbtablex.schema	Nome do esquema.
dbtablex.catalog	Catálogo associado à tabela atual.
filex.name	Nome do arquivo para arquivo de dados. <i>Observação:</i> A ordem dos campos no novo arquivo de entrada deve ser a mesma da ordem dos campos no fluxo originalmente publicado.
filex.path	Local do arquivo (nome do diretório).
filex.field_names_included	Para arquivos de formato de texto, a flag indicando se os nomes do campo são incluídos como a primeira linha do arquivo. (O valor é "Y" ou "N".)
filex.field_separator	Para arquivos de formato de texto, os caracteres usados para separar valores de campo.
filex.decimal_separator	Para arquivos de formato de texto, o caractere usado para a vírgula decimal.

Tabela 3. Parâmetros de execução (continuação)

Parâmetro	Propósito
options.angle_in_radians	Sinalização que indica se os radianos são usados como a unidade de medição nas expressões trigonométricas de CLEM. (O valor é "Y" ou "N".)
options.date_2digit_baseline	Define o século para as datas especificadas com anos de dois dígitos.
options.date_baseline	O ano de linha de base (sempre 1 de janeiro) usado pelas funções de data do CLEM que funcionam com uma única data.
options.time_rollover	Sinalização que indica se as diferenças de tempo negativo se referem ao passado.
options.decimal_separator	O caractere de vírgula decimal padrão nos arquivos de texto.
options.time_format	Formato de horário usando quando as sequências de caracteres são interpretadas como tempos pelas funções de tempo do CLEM.
options.date_format	Formato de dados usado quando as sequências de caracteres são interpretadas como datas pelas funções de data do CLEM.
options.timestamp_format	Formato usado ao ler os campos do registro de data e hora como sequências de caracteres das origens de dados ODBC.
paramx.value	Valor de parâmetro de fluxo. <i>Observação:</i> O nome do parâmetro, descrição e tipo de armazenamento estão incluídos para referência, mas não podem ser alterados. O valor deve ser válido para o tipo.

Incorporando o Editor de soluçõesIBM SPSS Modeler nos Aplicativos

Além do mecanismo de Tempo de Execução independente, o Editor de soluçõesIBM SPSS Modeler fornece uma biblioteca de programação de Tempo de Execução (CLEMRTL) que permite que outros programas controlem a execução de fluxo do Editor de soluçõesIBM SPSS Modeler. É possível chamar procedimentos CLEMRTL em programas clientes gravados em C e C + +. Para usar o CLEMRTL, é necessário incluir o arquivo de cabeçalho *clemrtl.h* (disponível na pasta .. \install\clemrtl\include) e certifique-se de vincular o arquivo de biblioteca apropriado para sua plataforma de desenvolvimento em seu aplicativo ao construí-lo. O arquivo *clemrtl.dll* está disponível na pasta ..\install\bin.

Qualquer arquivo de origem que refcrcie os procedimentos da biblioteca devem incluir o cabeçalho *clemrtl.h*. Esse arquivo de cabeçalho fornece protótipos C de ANSI para os procedimentos da biblioteca e define macros úteis. Isso não requer que outros cabeçalhos sejam incluídos além do que o seu programa requer. Para proteger contra conflitos de nomes, todos os tipos de bibliotecas e nomes de funções são iniciados com *clemrtl_* e todos os nomes de macros são prefixados com *CLEMRTL_*.

CLEMRTL produz relatórios contendo informações úteis que devem ser comunicadas de volta ao aplicativo de alguma maneira. O CLEMRTL fornece mecanismos para lidar com tais mensagens:

- O aplicativo pode recuperar detalhes do último erro utilizando a função *clemrtl_getErrorDetail()*.
- O aplicativo pode fornecer seu próprio procedimento de manipulação de relatórios usando a função *clemrtl_setReportHandler()*.

O sistema de Runtime possui um catálogo de mensagens localizadas para relatórios. Qualquer relatório transmitido para o aplicativo incluiria a sequência de mensagem localizada. Os relatórios também

incluem o código de relatório, para que o aplicativo possa optar por interpretar e apresentar a mensagem de maneira diferente. Para obter mensagens detalhadas (localizadas), o aplicativo precisa enviar o arquivo *messages.cfg* apropriado em um diretório *config*. Diferentes versões do *messages.cfg* para diferentes códigos de idioma podem ser localizadas no diretório <*installdir*>/*config*/<*locale*>, em que <*installdir*> é o diretório no qual você instalou o Editor de soluçõesIBM SPSS Modeler <*locale*> é o código de idioma desejado Seleccione a versão apropriada de *messages.cfg* e a inclua em um subdiretório *config* no diretório do projeto.

O sistema de Runtime não necessita de um ambiente especial ou configurações de registro para ser operado corretamente. As bibliotecas dependentes devem ser distribuídos com um aplicativo vinculado ao CLEMRTL. Esses arquivos estão incluídos no diretório <*installdir*>/*bin* .

Requisitos

A tabela a seguir mostra as plataformas e compiladores testados com a biblioteca de programação do Tempo de Execução.

Tabela 4. Plataformas e compiladores	
Plataforma	Compiladores sugeridos
Linux 64 bits	gcc 3.2.3, Intel C++ compiler 9.1.043
Windows	Visual Studio 2008 R2, Visual Studio 2015

Aplicativo Editor de soluçõesIBM SPSS Modeler Sample

Um aplicativo de amostra, *dlltest.c*, está incluído na pasta |clemtrtl|demo no diretório de instalação Editor de soluçõesIBM SPSS Modeler. Use o aplicativo de exemplo para ajudá-lo a começar a incorporar a biblioteca de Tempo de Execução em seus próprios aplicativos.

As informações usadas pelo aplicativo de exemplo estão incluídas na pasta |clemtrtl|demo; consulte o arquivo *README.txt* para obter instruções. A compilação de um aplicativo varia por plataforma, portanto, makefiles de exemplo também estão incluídos.

Capítulo 4. Instalando Tempo de execuçãoIBM SPSS Modeler Solution Publisher

Instalando o Tempo de Execução

Instalar o Tempo de execuçãoIBM SPSS Modeler Solution Publisher permite que você execute fluxos publicados. Os fluxos publicados são independentes do sistema operacional, dessa forma é possível executar qualquer fluxo publicado em qualquer plataforma suportada pelo Tempo de Execução.

A funcionalidade que permite a você publicar fluxos de IBM SPSS Modeler está instalada com IBM SPSS Modeler. Observe que os arquivos publicados devem ser exportados novamente com cada liberação do IBM SPSS Modeler. Por exemplo, um arquivo exportado usando uma versão anterior não funcionará com a versão atual de Tempo de execuçãoIBM SPSS Modeler Solution Publisher.

Instalando no Windows

A seção a seguir descreve o procedimento para a instalação do Tempo de execuçãoSPSS Modeler Solution Publisher no sistema operacional Windows.

1. Ative a instalação Editor de soluçõesIBM SPSS Modeler .
2. No assistente de instalação, clique em **Avançar** para iniciar.
3. Siga as instruções exibidas na tela. Para continuar, clique em **Avançar**.
4. Depois de ter especificado todas as opções, você estará pronto para instalar. Clique em **Instalar** para iniciar os arquivos de transferência.
5. Após todos os arquivos terem sido instalados, clique em **Concluir**.

Instalando no UNIX

As seções a seguir descrevem os requisitos adicionais do sistema e o procedimento para a instalação do Tempo de execuçãoIBM SPSS Modeler Solution Publisher no sistema operacional UNIX.

Requisitos Adicionais para UNIX

Se você estiver usando o Solution Publisher na mesma máquina que ServidorIBM SPSS Modeler, você deve garantir que os limites do kernel no sistema sejam suficientes para a operação de ServidorIBM SPSS Modeler. Os ulimits de dados, memória, arquivo e processos são particularmente importantes e devem ser configurados para ilimitados dentro do ambiente ServidorIBM SPSS Modeler . Para isso:

1. Inclua os seguintes comandos no `modelersrv.sh`:

```
ulimit -d unlimited  
ulimit -m unlimited  
ulimit -f unlimited  
ulimit -u unlimited
```

Além disso, configure o limite de pilha para o máximo permitido pelo seu sistema (`ulimit -s XXXX`), por exemplo:

```
ulimit -s 64000
```

2. Reinicie o ServidorIBM SPSS Modeler.

Você também precisa do utilitário de compressão de arquivos *gzip* e do *GNU cpio* instalado e no PATH para que o instalador possa descompactar os arquivos de instalação. Além disso, na máquina que está executando o Servidor do SPSS Modeler, você deve configurar o código de idioma para EN_US.UTF-8.

Procedimento de Instalação do UNIX

Estas instruções de instalação se aplicam ao Tempo de execução IBM SPSS Modeler Solution Publisher para UNIX. É necessário possuir permissões de leitura e gravação para o diretório de instalação de destino para efetuar logon com uma conta que possua permissões suficientes.

Nota: Você deve ter efetuado login como *raiz* para instalar.

1. A partir do pacote de instalação, abra o diretório *modelrun*.
2. Altere para o diretório de plataforma relevante.
3. Execute o script de instalação *.bin*. Por exemplo:

```
./modelersolutionpublisherhpiia64.bin -i console
```

4. Os detalhes da introdução são exibidos. Pressione Enter para continuar.
5. As informações sobre licença são exibidas. Leia a licença, digite 1 para aceitá-la e pressione Enter para continuar.
6. Uma lista com os idiomas disponíveis será exibida. Insira o número da versão do idioma que deseja instalar e pressione Enter para continuar.
7. Será solicitado a digitar o local de instalação. Para usar o diretório padrão, */usr/IBM/SPSS/ModelerSolutionPublisher<nn>* (onde <nn> é o número da versão), pressione Enter. O Tempo de Execução será instalado do diretório especificado.
8. Será solicitado que você confirme o local de instalação. Quando ele estiver correto, digite y e pressione Enter.
9. Um resumo da pré-instalação é exibido para confirmar suas entradas até o momento. Pressione Enter para continuar.
10. Uma mensagem é exibida para informar que a rotina de instalação está pronta para executar. Pressione Enter para continuar.
11. Uma barra de progresso é exibida enquanto a rotina de instalação é executada. Quando a instalação for concluída, pressione Enter para sair do instalador.

Configurando ODBC no UNIX

Para configurar o ODBC para iniciar com o Tempo de execução IBM SPSS Modeler Solution Publisher

Quando você pode se conectar com sucesso ao banco de dados a partir de Servidor IBM SPSS Modeler, você pode configurar uma instalação Tempo de execução IBM SPSS Modeler Solution Publisher no mesmo servidor referenciando o mesmo script *odbc.sh* do script de inicialização de Tempo de execução IBM SPSS Modeler Solution Publisher.

1. Edite o arquivo de script *modelerrun* em Tempo de execução IBM SPSS Modeler Solution Publisher para adicionar a linha a seguir imediatamente acima da última linha do script:

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

onde *odbc.sh_path* é o caminho completo para o arquivo *odbc.sh*. Por exemplo:

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

Nota: A sintaxe é importante aqui. Assegure-se de deixar um espaço entre o primeiro ponto e o caminho para o arquivo.

2. Salve o arquivo de script modelerrun.
3. Por padrão, o DataDirect Driver Manager não está configurado para o Tempo de execução IBM SPSS Modeler Solution Publisher para usar ODBC em sistemas UNIX. Para configurar o UNIX para carregar o DataDirect Driver Manager, insira os comandos a seguir (em que sp_install_dir é o diretório de instalação do Solution Publisher Runtime):

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

Resolvendo Problemas em uma Instalação

Assinatura Digital Inválida na Instalação

Os produtos IBM SPSS Modeler usam a certificação emitida pela IBM para assinatura digital. Em determinadas circunstâncias é possível ver o erro a seguir na tentativa de instalar os produtos ModeladorSPSS:

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

Todos os Usuários do Windows

Você verá essa mensagem se tentar instalar os produtos ModeladorSPSS em uma máquina que não tem conexão com a Internet e não tem o certificado correto instalado. Use o procedimento a seguir para corrigir esse problema.

1. Clique em **OK** para confirmar a mensagem.
2. Clique em **Cancelar** para sair do instalador.
3. Se a máquina na qual deseja instalar não tiver conexão com a Internet, execute a próxima etapa em uma máquina conectada à Internet e copie o arquivo .cer para a máquina na qual deseja instalar.
4. Acesse <https://support.symantec.com>, procure o **VeriSign Class 3 Primary Certification Authority - G5 root certificate** e faça o download. Salve-o como um arquivo .cer.
5. Clique duas vezes no arquivo .cer.
6. Na guia Geral, clique em **Instalar Certificado**.
7. Siga as instruções no Assistente de Importação de Certificado usando as opções padrão e clicando em **Concluir** no final.
8. Tente a instalação novamente.

Iniciando o Tempo de Execução

Quando tiver instalado o Tempo de Execução, será possível usá-lo para executar fluxos que foram publicados de IBM SPSS Modeler usando quaisquer um dos nós de exportação. Veja o tópico “[fluxos de publicação](#)” na [página 6](#) para obter mais informações. Inicie o Tempo de Execução da linha de comandos, com opções que indicam o nome do arquivo de imagem publicado e um arquivo de parâmetro opcional a ser executado.

Para executar o tempo de execução, digite a linha a seguir no prompt de comandos:

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

onde as entradas de comando são mostradas na tabela a seguir.

Tabela 5. Entradas de comando de Tempo de Execução

Entrada de Comando	Descrição
<parameter-file>	é o nome do arquivo do arquivo de parâmetro publicado (opcional).
<options>	é uma lista separada por vírgulas de pares de option=value especificando opções de execução para o Tempo de Execução. Observação: Você não deve deixar nenhum espaço em branco após uma vírgula.
<image file>	é o nome do arquivo do arquivo de imagem publicado (*.pim) a ser executado.

Observação: ao usar idiomas com caracteres multibyte, tais como chinês ou japonês em um ambiente UNIX, você deve especificar os códigos de idioma a serem usados. O exemplo a seguir mostra os códigos de idioma para chinês simplificado:

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

em que -p é a localização dos arquivos pim e par.

Veja Capítulo 3, “Executando fluxos publicados”, na página 9 para obter mais informações.

Alterando o diretório temporário

Algumas operações executadas pelo Tempo de execução IBM SPSS Modeler Solution Publisher pode requerer que arquivos temporários sejam criados. Por padrão, o Tempo de Execução usa o diretório temporário do sistema para criar arquivos temp. É possível alterar a localização do diretório temporário usando os passos a seguir.

Observação: é muito importante usar a sintaxe correta, sendo especialmente cuidadoso com espaços e o caminho do diretório.

Suponha que você esteja usando o Tempo de execução IBM SPSS Modeler Solution Publisher com os seguintes arquivos publicados—webtest.pim e webtest.par. Nessa situação, você deseja modificar a localização dos arquivos temporários para usar o diretório C:\published\temp. Para alterar a localização do diretório temp usado pelo executável:

1. Do diretório no qual os arquivos .pim e .par são salvos, execute o comando:

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

Isso irá direcionar o Tempo de Execução para usar o “C:\published\temp” como o diretório temp apenas para esse processo específico.

Desinstalando Editor de soluçõesIBM SPSS Modeler Runtime

A seção a seguir descreve os procedimentos para a desinstalação do Tempo de execução IBM SPSS Modeler Solution Publisher.

Procedimento de Desinstalação do Windows

1. No Painel de Controle do Windows, abra **Incluir ou Remover Programas**.
2. A partir da lista, selecione **IBM SPSS Modeler Solution Publisher < nn>**.
3. Clique em **Remover**.
4. Siga as instruções que aparecem na tela para remover o Tempo de Execução.

Observação: Você deve ter efetuado logon em seu computador com os privilégios de administrador para remover programas.

Procedimento de Desinstalação do UNIX

Remova o diretório de instalação, incluindo todos os arquivos instalados.

Capítulo 5. IBM SPSS Embedded Predictive Modeling API

O IBM SPSS Embedded Predictive Modeling API define uma interface Java simples para permitir que você integre os artefatos projetados do IBM SPSS Modeler com seus aplicativos em um mundo da moda integrado. A documentação completa no formato Javadoc está incluído com Editor de soluçõesIBM SPSS Modeler para os codificadores que estão usando a biblioteca.

Observe que as restrições a seguir ao usar a API de Embedded Predictive Modeling (EPM):

- **Origem exclusiva e rótulos de nó terminal são necessários.** A API EPM usa rótulos de nó. Embora os rótulos estejam sempre em um contexto de nó de Origem ou de Terminal, eles podem ser ambíguos.
- **Os super nós não são suportados como nós de origem ou nós terminais.** Para fornecer a origem comum para os super nós em todas as operações, eles não são suportados como nós de origem ou nós terminais
- **Restrições do nome de campo de entrada.** Alguns caracteres não podem ser usados nos nomes de campos.

Consulte as seguintes informações para uma visão geral da funcionalidade da API. Para obter detalhes adicionais, consulte o Javadoc incluído no produto.

Escoragem Integrada de Modelo Preditivo

Uso: usado pelo aplicativo para gerar a análise preditiva sob demanda conforme os novos dados entram. O aplicativo deve manipular modelos atualizados. Quaisquer nós de origem, bem como o nó terminal da ramificação de escoragem são substituídos pelas entradas e saídas do aplicativo neste modo de escoragem.

Ação: Prepara a ramificação de escoragem projetada por você em seu fluxo de IBM SPSS Modeler uma vez para chamadas repetidas que geram de maneira rápida e eficiente a análise preditiva dos dados de entrada gerenciados pelo aplicativo.

Atualização do Modelo Preditivo

Uso: Executa o treinamento inicial dos algoritmos modelo no design de fluxo usando dados históricos. Além disso, "atualiza" periodicamente os algoritmos de modelo preditivo configurado com os dados históricos mais atuais.

Ação: Executa todos os nós de construtores de modelo preditivo no fluxo de IBM SPSS Modeler, que treina o algoritmo de modelo preditivo conforme configurado usando técnicas de mineração de dados. Atualiza o nugget do aplicador de modelo vinculado a cada nó de construção de modelo. Parâmetros de fluxo e propriedades do nó/origem podem ser usados para implementar algum grau de dinâmica.

Avaliação do Modelo Preditivo

Uso: Usado por um aplicativo para comunicar a precisão, confiança ou outra avaliação qualitativa de um modelo preditivo treinado conforme projetado no fluxo de IBM SPSS Modeler. Estas informações são normalmente usadas para controlar a promoção de um modelo preditivo recentemente atualizado na escoragem no aplicativo. A razão primária que o Modelo Refresh retorna um novo objeto PredictiveModel é permitir a comparação de dados de avaliação para o modelo recém-atualizado com o original.

Ação: Executa todos os nós do construtor de documentos (principalmente os nós de saída não interativos nos paletes de Gráfico e Saída do IBM SPSS Modeler) no fluxo de IBM SPSS Modeler retornando os documentos de resultado. Parâmetros de fluxo e propriedades do nó/origem podem ser usados para implementar algum grau de dinâmica.

Execução de Fluxo do Modelo Preditivo

Uso: Todos os casos de uso que não se ajustam em "atualizar" "avaliação" ou nos dois modos de escoragem devem ser manipulados usando a execução de fluxo.

Ação: Execução conforme controlada pelas opções de fluxo do IBM SPSS Modeler. Seja no modo "todos os nós terminais" semelhante ao clicar no botão com a grande seta verde no Cliente IBM SPSS Modeler ou conforme codificado no script de nível de fluxo. Parâmetros de fluxo e propriedades do nó/origem podem ser usados para implementar algum grau de dinâmica.

Dados Externos de Escoragem do Modelo Preditivo

Uso: Comumente referido como um processo "em lote" devido à maneira em que um lote completo de dados de entrada é processado.

Ação: Executa a ramificação de escoragem conforme projetado. Entradas dos nós de origem de dados configurados e análise preditiva gerada são persistentes enquanto são controladas pelo nó terminal. Parâmetros de fluxo e propriedades do nó/origem podem ser usados para implementar algum grau de dinâmica.

Parâmetros de Fluxo do Modelo Preditivo e Especificação do Nó de Origem e Propriedades do Nó Terminal

Uso: Uma maneira comum é definir parâmetros que são referidos em expressões nas ramificações do fluxo. Outro ponto de controle é modificar determinadas propriedades-chave de um nó terminal ou de origem para alterar o caminho de arquivo usado, etc.

Ação: Determinados aspectos da execução de fluxo do IBM SPSS Modeler podem ser dinamicamente modificados por um aplicativo.

Nota: As mudanças do parâmetro devem ser feitas antes da criação do PredictiveScorer (consulte a descrição de escoragem integrada acima) e as propriedades do nó terminal e de objeto não possuem significado a este objeto.

Capítulo 6. Editor de soluções IBM SPSS Modeler

Referência da API da Biblioteca de Tempo de Execução

Introdução

A API CLEMRTL permite que você controle a execução dos fluxos publicados de seu aplicativo. Esta seção descreve as funções de API disponíveis.

A seguir estão os problemas gerais em uso na API:

- A API possui ligação C para máximo compatibilidade, porém a biblioteca possui dependências C++. Em algumas plataformas, isso pode significar que ela pode ser usada apenas com um vinculador compatível com C++.
- `clemrtl_initialise_ext()` deve ser chamado antes de quaisquer outras funções na biblioteca sejam usadas.
- O tipo `clemrtl_image_handle` é usado como um identificador de imagem.
- Cada função retorna um código indicador de status. Os valores do indicador de status são exibidos na tabela a seguir.

Tabela 6. Valores do indicador de status	
Resultado	Descrição
CLEMRTL_OK	Sucesso
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível
CLEMRTL_ERROR	Falha com informações adicionais sobre o erro disponível por meio da função <code>clemrtl_getErrorDetail()</code>

Visão Geral do Processo da API

A estrutura de tópicos geral de um aplicativo usando a API da seguinte maneira:

1. Inicialize a biblioteca usando `clemrtl_initialise_ext()`.
2. Verifique e altere as opções de execução usando `clemrtl_getOption()` e `clemrtl_setOption()`.
3. Abra uma imagem usando `clemrtl_openImage()` e receba uma manipulação de imagem.
4. Verifique e altere os parâmetros de imagem usando `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` e `clemrtl_setParameter()`.
5. Execute a imagem usando `clemrtl_execute()`. Se a mesma imagem deve ser executada várias vezes sem alterar parâmetros, use `clemrtl_prepare()` primeiro.
6. Feche a imagem usando `clemrtl_closeImage()`.
7. Para cancelar uma execução em andamento, use `clemrtl_interrupt()`.
8. Para recuperar informações sobre o último erro, use `clemrtl_getErrorDetail()`. Para receber mensagens de erro e outras mensagens de diagnóstico à medida em que elas chegam, use `clemrtl_setReportHandler()`.

A biblioteca é projetada para uso multiencadeado sob as seguintes condições:

- Funções globais, aquelas que não têm uma manipulação de imagem como um argumento, não podem ser chamadas simultaneamente.
- As funções locais de imagem, aquelas que têm uma manipulação de imagem como um argumento, não podem ser chamadas simultaneamente quando aplicadas na mesma manipulação de imagem, com a exceção do `clemrtl_interrupt()`.
- As funções de imagem local podem ser chamadas simultaneamente quando aplicadas a diferentes manipulações de imagem.

Em particular, isso significa que os encadeamentos separados podem preparar e executar imagens simultaneamente, desde que elas estejam usando diferentes manipulações de imagem. Diferentes manipulações de imagem podem se referir à mesma imagem (`clemrtl_openImage()` cria uma nova manipulação todas as vezes em que é chamada, mesmo para a mesma imagem), mas certifique-se ao executar diversas instâncias da mesma imagem para alterar os parâmetros de imagem para redirecionar o resultado, conforme necessário. Se duas imagens forem executadas simultaneamente, ambas tentam ser gravadas no mesmo arquivo de saída ou tabela de base de dados, os resultados serão imprevisíveis.

Funções de API

A seguir está uma lista completa das funções expostas na API.

inicializar

Essa função é descontinuada e é equivalente a

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

Novos programas devem chamar *initialise_ext*.

initialise_ext

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

Inicializa o Tempo de execução IBM SPSS Modeler Solution Publisher. *Esta função deve ser chamada antes de quaisquer outras funções API.*

Tabela 7. initialise_ext parameters

Parâmetro	Descrição
flags	Controla alguns aspectos do processo de inicialização. O valor é construído como um OR bit a bit das flags descritas abaixo.
arg_count	O número de argumentos de inicialização adicional. Deve ser inferior ou igual ao comprimento da matriz de args.
args	Argumentos de inicialização adicionais. Os argumentos são descritos pelo tipo: <code>typedef struct _clemrtl_init_arg {</code> <code> const char* name;</code> <code> const char* value;</code> <code>} clemrtl_init_arg;</code>

O campo name é o nome de uma propriedade de configuração e o campo value é o seu valor. As propriedades de configuração mostradas na tabela a seguir são reconhecidas.

Tabela 8. Propriedades de Configuração	
Nome	Valor
installation_directory	O caminho completo para a pasta de instalação do Editor de soluções IBM SPSS Modeler. A pasta de instalação é a que contém as pastas bin e config.

Retorna um dos códigos de erro mostrados na tabela a seguir.

Tabela 9. Códigos de Erros	
Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha ao inicializar, sem detalhes adicionais disponíveis.

O parâmetro flags fornece controle sobre determinados aspectos do processo de inicialização. Um valor de 0 (ou CLEMRTL_INIT_DEFAULTS) especifica o comportamento padrão da seguinte maneira:

- Configure o fuso horário padrão chamando tzset().
- Configure o código de idioma LC_NUMERIC para “C”.
- Configure o novo manipulador global para emitir uma exceção quando a memória estiver esgotada.

Este comportamento pode ser modificado especificando combinações das flags mostradas na tabela a seguir.

Tabela 10. Sinalizações	
Sinalização	Descrição
CLEMRTL_INIT_NO_TZ	Não configurar o fuso horário.
CLEMRTL_INIT_NO_LOCALE	Não altere o código de idioma.
CLEMRTL_INIT_NO_NEW_HANDLER	Não configure o novo manipulador.
CLEMRTL_INIT_LOCAL_NEW_HANDLER	Localize o novo manipulador dentro de cada chamada API.

A biblioteca requer que o código de idioma LC_NUMERIC deve ser configurado para “C” para converter adequadamente entre números e sequências numéricas (o formato numérico IBM SPSS Modeler usa as regras do código de idioma “C”). Isso inclui a conversão de números lidos e gravados em arquivos de texto.

- Especifique NO_LOCALE se o seu aplicativo depender do código de idioma LC_NUMERIC e não requerer a conversão entre números e sequências de caracteres.
- NO_NEW_HANDLER e LOCAL_NEW_HANDLER são mutuamente exclusivos. A biblioteca requer que new lance uma exceção quando a memória se esgotar.
- Especifique NO_NEW_HANDLER se o seu aplicativo configurar seu próprio novo manipulador ou se você estiver certo de que o sistema de Tempo de Execução/compilador emite uma exceção por padrão.

Especifique LOCAL_NEW_HANDLER apenas se o seu aplicativo depender de new retornando um ponteiro nulo quando a memória estiver esgotada. *Observação:* Não é seguro para uso em um ambiente multiencadeado.

getOption

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

Recupera o valor de uma opção de execução.

Tabela 11. parâmetros de <i>getOption</i>	
Parâmetro	Descrição
name	O nome da opção.
value	Um buffer para receber o valor da opção.
value_size	O tamanho do buffer de valor.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 12. Códigos de status	
Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se o nome da opção não for um dos nomes de opção de execução listados acima, por exemplo, `memory_usage` ou se o buffer de valor não for grande o suficiente para receber o valor incluindo um caractere nulo de finalização. Os valores de opção são sempre sequências de caracteres, mesmo quando a interpretação for numérica. Consulte o [Capítulo 3, “Executando fluxos publicados”, na página 9](#) para obter mais informações.

setOption

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

Define o valor de uma opção de execução.

Tabela 13. parâmetros de <i>setOption</i>	
Parâmetro	Descrição
name	O nome da opção.
value	O valor da opção.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 14. Códigos de status	
Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se o nome da opção não for um dos nomes de opção de execução listados acima, por exemplo, `memory_usage`. Os valores de opção são sempre sequências de caracteres, mesmo quando a interpretação for numérica.

openImage

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

Abre uma imagem de fluxo publicado.

Tabela 15. parâmetros de openImage	
Parâmetro	Descrição
<code>image_file</code>	O nome do arquivo de imagem.
<code>param_file</code>	O nome do arquivo de parâmetro.
<code>handle</code>	Recebe a manipulação de imagem.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 16. Códigos de status	
Resultado	Descrição
<code>CLEMRTL_OK</code>	Sucesso.
<code>CLEMRTL_FAIL</code>	Falha com nenhum detalhe adicional disponível.
<code>CLEMRTL_ERROR</code>	Falha com detalhes adicionais disponíveis.

O arquivo de parâmetro pode ser NULL se um arquivo de parâmetro separado não for necessário. A operação falhará se o arquivo de imagem não estiver especificado, se o nome do arquivo for inválido ou se o conteúdo for ilegível. Se a operação for bem-sucedida, a manipulação de imagem retornada poderá ser usada para identificar a instância de imagem nas chamadas API subsequentes.

closeImage

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

Fecha uma manipulação de imagem e libera seus recursos.

Tabela 17. parâmetros de closeImage	
Parâmetro	Descrição
<code>handle</code>	A manipulação de imagem.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 18. Códigos de status	
Resultado	Descrição
<code>CLEMRTL_OK</code>	Sucesso.
<code>CLEMRTL_FAIL</code>	Falha com nenhum detalhe adicional disponível.
<code>CLEMRTL_ERROR</code>	Falha com detalhes adicionais disponíveis.

A operação falhará se outra chamada API estiver em progresso na imagem, especificamente se a imagem ainda estiver em execução. Se a operação for bem-sucedida, a imagem será encerrada e a manipulação não poderá ser usada em nenhuma chamada API subsequente, exceto em outra chamada de `clemrtl_closeImage()`, que não tem nenhum efeito.

enumerateParameters

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

Aplica um procedimento de retorno de chamada para cada nome e valor de parâmetro de imagem.

Tabela 19. *enumerateParameters parameters*

Parâmetro	Descrição
handle	A manipulação de imagem.
proc	O procedimento de retorno de chamada do parâmetro.
data	Dados especificados pelo usuário para o retorno de chamada.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 20. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

O procedimento de retorno de chamada possui o seguinte tipo:

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

Tabela 21. Parâmetros de procedimento de retorno de chamada

Parâmetro	Descrição
data	Dados especificados pelo usuário aprovados para <code>clemrtl_enumerateParameters()</code> .
name	O nome do parâmetro.
value	O valor do parâmetro.

O procedimento é aplicado exatamente uma vez para cada parâmetro de imagem em uma ordem arbitrária.

getParameter

```
int clemrtl_getParameter(
    clemrtl_image_handle handle,
    const char* name,
```

```
char* value,
int value_size);
```

Recupera o valor de um parâmetro de imagem.

Tabela 22. parâmetros de getParameter

Parâmetro	Descrição
handle	A manipulação de imagem.
name	O nome do parâmetro.
value	Um buffer para receber o valor de parâmetro.
value_size	Tamanho do buffer de valor.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 23. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se o nome do parâmetro não corresponder ao nome de nenhum parâmetro na imagem ou se o buffer de valor não for grande o suficiente para receber o valor de parâmetro incluindo um caractere nulo de finalização. Os nomes do parâmetro usam o formato name.attribute descrito acima, por exemplo, file0.name e os valores de parâmetro são sempre sequências de caracteres, mesmo quando a interpretação é numérica.

setParameter

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

Define o valor de um parâmetro de imagem.

Tabela 24. parâmetros de setParameter

Parâmetro	Descrição
handle	A manipulação de imagem.
name	O nome do parâmetro.
value	O valor do parâmetro.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 25. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se a manipulação de imagem for inválida ou se o nome do parâmetro não corresponder ao nome de nenhum parâmetro na imagem. Os nomes do parâmetro usam o formato `name.attribute` descrito acima, por exemplo, `file0.name` e os valores de parâmetro são sempre sequências de caracteres, mesmo quando a interpretação é numérica.

getFieldCount

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

Retorna o número de campos em uma origem de entrada ou uma resposta de saída.

Tabela 26. parâmetros de getFieldType	
Parâmetro	Descrição
handle	A manipulação de imagem.
key	O nome da entrada ou saída a ser examinada, conforme usado no arquivo de parâmetros. A chave pode se referir a um arquivo ou um banco de dados.
field_count	Recebe o número de campos.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 27. Códigos de status	
Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

getFieldTypes

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

Retorna os tipos de campo para uma origem de entrada ou um destino de saída.

Tabela 28. parâmetros de getFieldTypes	
Parâmetro	Descrição
handle	A manipulação de imagem.
key	O nome da entrada ou saída a ser examinada, conforme usado no arquivo de parâmetros. A chave pode se referir a um arquivo ou um banco de dados.
field_count	O número de campos a serem examinados. Deve ser inferior ou igual ao comprimento da matriz do <code>field_types</code> .

Tabela 28. parâmetros de getFieldTypes (continuação)

Parâmetro	Descrição
field_types	Uma matriz de comprimento de pelo menos field_count que recebe os tipos de campo. Os tipos dos primeiros campos field_count na entrada ou saída são copiados na matriz. Se field_count é maior que o número real dos campos, os elementos extra na matriz são deixados indefinidos. Os valores para os tipos de campo são listados na tabela Tipos de Dados abaixo.

Tipos de Dados

O field_types deve ser um dos tipos de dados mostrados na tabela a seguir.

Tabela 29. Tipos de dados

Tipo	Interpretação	Declarador "C" Típico
STRING	Sequência de caracteres terminada em zero UTF-8.	const char*
NÚMERO INTEIRO	Número inteiro assinado de 32 bits.	int
LONG	Número inteiro assinado de 64 bits.	longo longo
REAL	Ponto flutuante de 64 bits.	duplo
Horário	Número inteiro assinado de 64 bits (segundos desde a meia-noite).	longo longo
DATA	Número inteiro assinado de 64 bits (segundos desde a meia-noite em 01/01/1970).	longo longo
TIMESTAMP	Número inteiro assinado de 64 bits (segundos desde a meia-noite em 01/01/1970).	longo longo

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 30. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

setAlternativeInput

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

Substitui uma origem de entrada de arquivo com uma origem de entrada alternativa.

Tabela 31. parâmetros de setAlternativeInput

Parâmetro	Descrição
handle	A manipulação de imagem.
key	O nome da origem de entrada original conforme usado no arquivo de parâmetros. A fonte de entrada deve ser um arquivo, de modo que a chave sempre terá o formato "fileN" para algum número inteiro $N \geq 0$.
field_count	O número de campos na entrada. O valor deve corresponder exatamente ao número de campos na entrada do arquivo original ou a chamada falhará.
field_types	Uma matriz de tipos de comprimento de pelo menos field_count. O valor field_types[i] é o tipo do campo i e deve ser um dos valores listados na tabela Tipos de Dados abaixo. Os tipos devem ser compatíveis com aqueles da entrada de arquivo original ou a chamada falhará.
iterator	A função que produz os dados de entrada alternativa. A função é aplicada ao seu argumento, da seguinte maneira: <pre>void** row = iterator(arg);</pre> A função é chamada durante a execução (dentro de uma chamada para clemrtl_execute) e é chamada uma vez para cada registro de entrada. Um valor de retorno de NULL indica o término de entrada em cujo caso a função não é chamada novamente e a execução será eventualmente encerrada. Caso contrário, o resultado é uma matriz de dados de comprimento de pelo menos field_count em que row[i] fornece o valor do campo i. Um valor pode ser NULL, caso contrário, ele deve ser um ponteiro para um datum cujo tipo é determinado pelo field_types[i] correspondente. Os ponteiros devem permanecer válidos até a próxima chamada do agente iterativo ou até o término da execução, caso a execução seja encerrada prematuramente.
arg	Um argumento opaco que é transmitido para o agente iterativo em cada chamada.

Tipos de Dados

O field_types deve ser de uma lista restrita de tipos. Veja o tópico “[getFieldTypes](#)” na página 30 para obter mais informações.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 32. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se os tipos de campos não forem compatíveis com aqueles da entrada de arquivo original ou se a contagem de campo não corresponder exatamente ao número de campos na entrada de arquivo original.

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);
```

Substitui uma resposta de saída de arquivo com uma resposta de saída alternativa.

<i>Tabela 33. parâmetros de setAlternativeOutput</i>	
Parâmetro	Descrição
handle	A manipulação de imagem.
key	O nome da resposta de saída original conforme usado no arquivo de parâmetros. O destino de saída deve ser um arquivo, portanto, a chave sempre terá o formato "fileN" para algum número inteiro $N \geq 0$.
field_count	O número de campos na saída. O valor deve corresponder exatamente ao número de campos no resultado de arquivo original ou a chamada falhará.
field_types	Uma matriz de tipos de comprimento de pelo menos <code>field_count</code> . O valor <code>field_types[i]</code> é o tipo do campo i e deve ser um dos valores listados na tabela Tipos de Dados abaixo. Os tipos devem ser compatíveis com aqueles do arquivo original do resultado de arquivo original ou a chamada falhará.
iterator	<p>Uma função que consome o resultado da imagem. A função é aplicada ao seu argumento e para uma linha de dados, da seguinte maneira:</p> <pre>iterator(arg, row);</pre> <p>A função é chamada durante a execução (dentro de uma chamada para <code>clemrtl_execute</code>) e é chamada uma vez para cada linha de resultado produzida pela imagem. Um valor <code>row</code> de <code>NULL</code> indica o fim da saída após o qual a função não é chamada novamente; um aplicativo não deve contar com esta chamada final e deve flush e fechar qualquer recurso externo, etc. quando a execução tiver finalizado. Caso contrário, o <code>row</code> é uma matriz de dados de comprimento pelo menos <code>field_count</code> onde <code>row[i]</code> fornece o valor do $i^{\text{ésimo}}$ resultado. Um valor pode ser <code>NULL</code>, caso contrário, ele deve ser um ponteiro para um <code>datum</code> cujo tipo é determinado pelo <code>field_types[i]</code> correspondente. A função não deve copiar nenhum valor dos dados necessários, porque a memória pode não ser preservada após a chamada ter sido retornada.</p>

Tabela 33. parâmetros de setAlternativeOutput (continuação)

Parâmetro	Descrição
arg	Um argumento opaco que é transmitido para o agente iterativo em cada chamada.

Tipos de Dados

O field_types deve ser de uma lista restrita de tipos. Veja o tópico “[getFieldTypes](#)” na página 30 para obter mais informações.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 34. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_FAIL	Falha com nenhum detalhe adicional disponível.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se os tipos de campo não forem compatíveis com aqueles do resultado de arquivo original ou se a contagem de campo não corresponder exatamente ao número de campos no resultado do arquivo original.

executar

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

Executa uma imagem.

Tabela 35. parâmetros de execução

Parâmetro	Descrição
handle	A manipulação de imagem.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 36. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

Se a imagem não tiver sido preparada, primeiro, ela será preparada usando os valores de parâmetros atuais. A operação falhará se a manipulação de imagem for inválida ou se um erro ocorrer durante a preparação ou execução. A chamada não é retornada até que a execução esteja concluída.

preparar

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

Prepara uma imagem para execução.

Tabela 37. parâmetros de preparação

Parâmetro	Descrição
handle	A manipulação de imagem.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 38. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação falhará se a manipulação de imagem for inválida ou se um erro ocorrer durante a preparação.

Uma imagem deve ser preparada antes de ser executada. Preparar uma imagem congela os valores de parâmetro na imagem. `clemrtl_execute()` prepara uma imagem automaticamente em cada chamada se ainda não tiver sido preparada; usar `clemrtl_prepare()` para preparar uma imagem é útil se a imagem for executada diversas vezes com os mesmos valores de parâmetro e isso pode melhorar significativamente o desempenho. Depois que a imagem tiver sido preparada, as mudanças subsequentes para os valores de parâmetro serão ignoradas; use `clemrtl_prepare()` novamente para atualizar a imagem com os novos valores de parâmetro.

interrupção

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

Finaliza a execução em progresso em uma imagem.

Tabela 39. parâmetros de interrupção

Parâmetro	Descrição
handle	A manipulação de imagem.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 40. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

A operação não terá efeito se a manipulação de imagem for inválida ou se a imagem não estiver em execução.

Esta função é segura para ser chamada simultaneamente com outra chamada API na mesma manipulação de imagem.

getErrorDetail

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

Recupera informações detalhadas sobre o último erro que ocorreu em uma imagem.

Tabela 41. parâmetros de getErrorDetail	
Parâmetro	Descrição
handle	A manipulação de imagem.
severity	Recebe o código de severidade como um caractere único: Informações-I Aviso-W Erro-E Erro no sistema-X
code	Recebe o número do erro.
text	Um buffer para receber o texto da mensagem.
text_size	Tamanho do buffer de texto.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 42. Códigos de status	
Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

Os resultados não serão confiáveis se a manipulação de imagem for inválida. O texto da mensagem será truncado, se necessário, para se ajustar no buffer de texto.

Se uma chamada API falhar com CLEMRTL_ERROR e o erro não estiver relacionado a uma manipulação de imagem particular, aprovar 0 como a manipulação de imagem recuperará detalhes do último erro não específico de imagem, mas não será confiável em um ambiente multiencadeado.

setReportHandler

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

Instala um procedimento de retorno de chamada de relatório para uma imagem.

Tabela 43. parâmetros de setReportHandler	
Parâmetro	Descrição
handle	A manipulação de imagem.
proc	O procedimento de retorno de chamada de relatório.
data	Dados especificados pelo usuário para o retorno de chamada.

Retorna um dos códigos de status mostrados na tabela a seguir.

Tabela 44. Códigos de status

Resultado	Descrição
CLEMRTL_OK	Sucesso.
CLEMRTL_ERROR	Falha com detalhes adicionais disponíveis.

O procedimento de retorno de chamada possui o seguinte tipo:

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

Tabela 45. Parâmetros de procedimento de retorno de chamada

Parâmetro	Descrição
data	Dados especificados pelo usuário aprovados para <code>clemrtl_setReportHandler()</code> .
severity	O código de severidade como um caractere único: Informações-I Aviso-W Erro-E Erro no sistema-X
code	O número da mensagem.
text	O texto da mensagem.

O procedimento é aplicado a cada mensagem conforme ela chega. Configurar o procedimento para NULL remove qualquer manipulador existente instalado na imagem.

Avisos

Estas informações foram desenvolvidas para os produtos e serviços oferecidos nos EUA. Este material pode estar disponível na IBM em outros idiomas. No entanto, pode ser necessário possuir uma cópia do produto ou da versão do produto no mesmo idioma para acessá-lo.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não lhe garante direito algum sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

*Gerência de Relações Comerciais e Industriais da IBM Brasil
IBM Corporation
Botafogo
Rio de Janeiro, RJ
EUA*

Para pedidos de licença relacionados a informações de Conjunto de Caracteres de Byte Duplo (DBCS), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE MERCADO OU DE ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nestes documentos ou Web sites não fazem parte dos materiais deste produto IBM e a utilização destes documentos ou Web sites é de inteira responsabilidade do Cliente.

A IBM pode usar ou distribuir quaisquer informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

*Gerência de Relações Comerciais e Industriais da IBM Brasil
IBM Corporation*

*Botafogo
Rio de Janeiro, RJ
EUA*

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Os exemplos de clientes e dados de desempenho citados são apresentados com propósitos meramente ilustrativos. Os resultados reais de desempenho podem variar, dependendo das configurações e condições operacionais específicas.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

As declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos esses nomes são fictícios e qualquer semelhança com pessoas ou empresas reais é mera coincidência.

Marcas comerciais

IBM, o logotipo IBM e ibm.com são marcas comerciais ou marcas registradas da International Business Machines Corp., registradas em várias jurisdições no mundo todo. Outros nomes de produto e de serviço podem ser marcas registradas da IBM ou de outras empresas. Uma lista atual de marcas registradas IBM está disponível na web em "Informações de copyright e marca registrada" em www.ibm.com/legal/copytrade.shtml.

Adobe, o logotipo Adobe, PostScript e o logotipo PostScript são marcas registradas ou marcas comerciais da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

Intel, o logotipo Intel, Intel Inside, o logotipo Intel Inside, Intel Centrino, o logotipo do Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium são marcas comerciais ou marcas registradas da Intel Corporation ou suas subsidiárias nos Estados Unidos e em outros países.

Linux é uma marca registrada da Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e/ou em outros países.

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.

Termos e condições para documentação do produto

As permissões para uso destas publicações são concedidas sujeitas aos seguintes termos e condições.

Aplicabilidade

Estes termos e condições estão em adição aos termos de uso para o website da IBM.

Uso pessoal

É possível reproduzir estas publicações para seu uso pessoal não comercial, desde que todos os avisos do proprietário sejam preservados. Você pode não distribuir, exibir ou fazer trabalhos derivados dessas publicações, ou de qualquer parte delas, sem o consentimento expresso da IBM.

Uso Comercial

O Cliente pode reproduzir, distribuir e exibir estas publicações unicamente dentro da empresa, desde que todos os avisos do proprietário sejam preservados. Você pode não fazer trabalhos derivados dessas publicações, ou reproduzir, distribuir ou exibir estas publicações ou qualquer parte delas fora de sua empresa, sem o consentimento expresso da IBM.

Direitos

Exceto quando expressamente concedido nesta permissão, nenhuma outra permissão, licença ou direito é concedido, seja de maneira expressa ou implícita, para as publicações ou quaisquer informações, dados, software ou outras propriedades intelectuais aqui contidas.

A IBM reserva-se o direito de retirar as permissões concedidas aqui sempre que, a seu critério, o uso das publicações seja prejudicial a seus interesses ou, conforme determinado pela IBM, as instruções acima não estejam sendo seguidas corretamente.

O Cliente não pode fazer download, exportar ou re-exportar estas informações, exceto se estiver em conformidade total com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação dos Estados Unidos.

A IBM NÃO FAZ QUALQUER TIPO DE GARANTIA QUANTO AO CONTEÚDO DESTAS PUBLICAÇÕES. AS PUBLICAÇÕES SÃO FORNECIDAS "NO ESTADO EM QUE SE ENCONTRAM" E SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, NÃO INFRAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Índice remissivo

A

API do
 códigos de erro [23](#)
 códigos de status [23](#)
 contagem de campo [30](#)
 entrada alternativa [31](#)
 opções de execução [26](#)
 referência [23](#)
 saída alternativo [33](#)
 setReportHandler [36](#)
 tipos de campo [30](#)
 visão geral [12, 23](#)
arquivo .par [6](#)
arquivo .pim [6](#)

B

Biblioteca de programação do Tempo de Execução (CLEMRTL) [12](#)

C

closeImage
 Referência de função da API [27](#)
 Visão geral de processo da API [23](#)
códigos de erro [23](#)
códigos de status [23](#)

D

desinstalando
 IBM SPSS Modeler Solution Publisher Runtime [18](#)
diretório temporário
 IBM SPSS Modeler Solution Publisher Runtime [18](#)
Documentação [3](#)

E

enumerateParameters
 Referência de função da API [28](#)
 Visão geral de processo da API [23](#)
EPM
 API do [21](#)
executando fluxos
 usando o IBM SPSS Modeler Solution Publisher [9, 10](#)
executar
 Referência de função da API [34](#)
 Visão geral de processo da API [23](#)
exemplos
 Guia de Aplicativos [3](#)
 visão geral [4](#)
exemplos de aplicativos [3](#)

F

fechando
 fluxos [27](#)
fluxos de publicação
 IBM SPSS Modeler Solution Publisher [5, 6](#)
funções de API
 closeImage [23, 27](#)
 enumerateParameters [23, 28](#)
 executar [23, 34](#)
 getErrorDetail [23, 35](#)
 getFieldCount [23, 30](#)
 getFieldTypes [23, 30](#)
 getOption [23, 26](#)
 getParameter [23, 28](#)
 inicialização [23](#)
 inicializando [24](#)
 interrupção [23, 35](#)
 openImage [23, 27](#)
 preparar [23, 34](#)
 setAlternativeInput [23, 31](#)
 setAlternativeOutput [23, 33](#)
 setLogFile [23](#)
 setOption [23, 26](#)
 setParameter [23, 29](#)
 setReportHandler [23](#)
fuso horário
 configurar utilizando a API [24](#)

G

getErrorDetail
 Referência de função da API [35](#)
 Visão geral de processo da API [23](#)
getFieldCount
 Referência de função da API [30](#)
 Visão geral de processo da API [23](#)
getFieldTypes
 Referência de função da API [30](#)
 Visão geral de processo da API [23](#)
getOption
 Referência de função da API [26](#)
 Visão geral de processo da API [23](#)
getParameter
 Referência de função da API [28](#)
 Visão geral de processo da API [23](#)

I

IBM SPSS Embeddable Predictive Analytics
 API do [21](#)
IBM SPSS Modeler
 Documentação [3](#)
IBM SPSS Modeler Server [1](#)
IBM SPSS Modeler Solution Publisher
 integrando a aplicativos [12](#)
 visão geral [5](#)

IBM SPSS Modeler Solution Publisher Runtime

- desinstalando [18](#)
- diretório temporário [18](#)
- iniciando [17](#)
- instalando [15](#)
- requisitos do sistema [15](#)
- implementação [5, 21](#)
- inicializar
 - Referência de função da API [24](#)
 - sinalizadores [24](#)
 - Visão geral de processo da API [23](#)
- initialise_ext
 - Referência de função da API [24](#)
 - sinalizadores [24](#)
- interrupção
 - Referência de função da API [35](#)
 - Visão geral de processo da API [23](#)

L

- localidade
 - configurar utilizando a API [24](#)

N

- Nó de Publicação [21](#)
- Nó do IBM SPSS Modeler Solution Publisher [21](#)
- nós de saída [5, 21](#)

O

- openImage
 - Referência de função da API [27](#)
 - Visão geral de processo da API [23](#)

P

- Parâmetros
 - Arquivos de parâmetro de API [27](#)
 - Parâmetros da imagem de API [28, 29](#)
 - parâmetros para execução de fluxo [10](#)
 - Procedimentos de retorno de chamada de API [28](#)
- preparar
 - Referência de função da API [34](#)
 - Visão geral de processo da API [23](#)

R

- relatório de erros
 - Referência de função da API [35, 36](#)

S

- setAlternativeInput
 - Referência de função da API [31](#)
 - Visão geral de processo da API [23](#)
- setAlternativeOutput
 - Referência de função da API [33](#)
 - Visão geral de processo da API [23](#)
- setLogFile
 - Visão geral de processo da API [23](#)
- setOption

setOption (*continuação*)

- Referência de função da API [26](#)
- Visão geral de processo da API [23](#)
- setParameter
 - Referência de função da API [29](#)
 - Visão geral de processo da API [23](#)
- setReportHandler
 - Referência de função da API [36](#)
 - Visão geral de processo da API [23](#)

T

- Tempo de execução
 - desinstalando [18](#)
 - diretório temporário [18](#)
 - iniciando [17](#)
 - Instalação do UNIX [15](#)
 - Instalação no Windows [15](#)
 - opções [9](#)

U

- UNIX
 - instalando o IBM SPSS Modeler Solution Publisher Runtime [15](#)

V

- vaga
 - fluxos [27](#)

IBM.[®]