

*IBM SPSS Modeler 18.1.1 Solution  
Publisher*

**IBM**

**Uwaga**

Przed skorzystaniem z niniejszych informacji oraz produktu, którego one dotyczą, należy zapoznać się z informacjami zamieszczonymi w sekcji “Uwagi” na stronie 37.

**Informacje o produkcie**

Niniejsze wydanie publikacji dotyczy wersji 18, wydania 1, modyfikacji 1 produktu IBM SPSS Modeler oraz wszystkich następnych wydań i modyfikacji do czasu, aż w kolejnym wydaniu publikacji zostanie zawarta informacja o stosownej zmianie.

---

## Spis treści

**Przedmowa . . . . . v**

### **Rozdział 1. O programie IBM SPSS**

**Modeler . . . . . 1**

Produkty IBM SPSS Modeler . . . . . 1

IBM SPSS Modeler . . . . . 1

IBM SPSS Modeler Server . . . . . 1

IBM SPSS Modeler Administration Console . . . . . 2

IBM SPSS Modeler Batch . . . . . 2

IBM SPSS Modeler Solution Publisher . . . . . 2

IBM SPSS Modeler Server Adapters for IBM SPSS

Collaboration and Deployment Services . . . . . 2

Wydania programu IBM SPSS Modeler . . . . . 2

Dokumentacja . . . . . 3

Dokumentacja SPSS Modeler Professional . . . . . 3

Dokumentacja SPSS Modeler Premium . . . . . 4

Przykłady zastosowań . . . . . 4

Folder Demos . . . . . 4

Monitorowanie wykorzystania licencji . . . . . 4

### **Rozdział 2. IBM SPSS Modeler Solution**

**Publisher . . . . . 7**

Jak działa oprogramowanie IBM SPSS Modeler Solution

Publisher . . . . . 7

Publikowanie strumieni . . . . . 8

### **Rozdział 3. Wykonywanie opublikowanych strumieni . . . . . 11**

Używanie pliku parametrów do dostosowania wykonywania strumienia . . . . . 12

Osadzanie produktu IBM SPSS Modeler Solution

Publisher w aplikacjach . . . . . 13

Przykładowa aplikacja IBM SPSS Modeler Solution

Publisher . . . . . 14

### **Rozdział 4. Instalowanie produktu IBM SPSS Modeler Solution Publisher**

**Runtime . . . . . 15**

Instalowanie środowiska wykonawczego . . . . . 15

Instalowanie w systemie Windows . . . . . 15

Instalowanie w systemie UNIX . . . . . 15

Rozwiązywanie problemów i instalacja . . . . . 16

Uruchamianie środowiska wykonawczego . . . . . 17

Zmiana położenia katalogu tymczasowego . . . . . 17

Deinstalacja produktu IBM SPSS Modeler Solution

Publisher Runtime . . . . . 18

Procedura deinstalacji w systemie Windows . . . . . 18

Procedura deinstalacji w systemie UNIX . . . . . 18

### **Rozdział 5. IBM SPSS API EPM (osadzone modelowanie predykcyjne) . 19**

### **Rozdział 6. Informacje o bibliotece środowiska wykonawczego API IBM SPSS Modeler Solution Publisher . . . 21**

Wstęp . . . . . 21

Przegląd procesu API . . . . . 21

Funkcje API . . . . . 22

**Uwagi . . . . . 37**

Znaki towarowe . . . . . 38

Warunki dotyczące dokumentacji produktu . . . . . 39

**Indeks . . . . . 41**



---

## Przedmowa

IBM® SPSS Modeler to oferowane przez IBM Corp. zaawansowane środowisko eksploracji danych. SPSS Modeler pomaga przedsiębiorstwom i instytucjom w rozwijaniu relacji z klientami i obywatelami w oparciu o pogłębioną interpretację dostępnych danych. Organizacje korzystają z wiedzy uzyskanej dzięki programowi SPSS Modeler w bardzo szerokim spektrum zastosowań, m.in. do zatrzymywania najbardziej wartościowych klientów, określania możliwości sprzedaży wiązanej, przyciągania nowych klientów, wykrywania oszustw, ograniczania ryzyka i podnoszenia jakości usług publicznych.

Interfejs graficzny produktu SPSS Modeler zachęca użytkowników, aby wykorzystywali specjalistyczną wiedzę, dzięki której możliwe będzie opracowanie bardziej wydajnych modeli predykcyjnych i skrócenie czasu potrzebnego do uzyskania rozwiązania. SPSS Modeler oferuje wiele technik modelowania, takich jak predykcja, klasyfikacja, segmentacja i algorytmy do wykrywania związków. Po utworzeniu modeli program IBM SPSS Modeler Solution Publisher umożliwia udostępnienie ich osobom podejmującym decyzje w całym przedsiębiorstwie lub zapisanie w bazie danych.

## Informacje o programie IBM Business Analytics

Oprogramowanie IBM Business Analytics dostarcza kompletne, spójne i dokładne informacje, na których mogą polegać osoby decyzyjne chcące polepszyć wyniki biznesowe. Wszechstronne portfolio obejmujące moduły: analiza biznesowa, analiza prognostyczna, zarządzanie wynikami i strategiami finansowymi oraz aplikacje analityczne, zapewnia jasny, natychmiastowy i pozwalający na podjęcie działań wgląd w bieżące wyniki oraz daje możliwość przewidywania przyszłych wyników. W połączeniu z licznymi rozwiązaniami branżowymi, sprawdzonymi praktykami i profesjonalnymi usługami, organizacje o różnych rozmiarach mogą wspomagać najwyższą produktywność, w sposób pewny zautomatyzować decyzje i uzyskać lepsze wyniki.

Oprogramowanie IBM SPSS Predictive Analytics będące częścią tego portfolio wspomaga organizacje w zakresie przewidywania przyszłych zdarzeń oraz proaktywnie wpływać na na ten wgląd z korzyścią dla wyników finansowych. Klienci komercyjni, rządowi i uczelnie na całym świecie polegają na technologii IBM SPSS zapewniającej przewagę konkurencyjną, dzięki której przyciągają, zatrzymują i pozyskują nowych klientów, walcząc z nieuczciwością i ograniczając ryzyko. Wdrażając oprogramowanie IBM SPSS do swojej codziennej działalności, organizacje stają się przewidującymi przedsiębiorstwami, zdolnymi do zarządzania i automatyzacji decyzji w celu realizacji celów biznesowych i osiągnięcia mierzalnej przewagi konkurencyjnej. W celu uzyskania dalszych informacji lub skontaktowania się z przedstawicielem, proszę wejść na stronę <http://www.ibm.com/spss>.

## Wsparcie techniczne

Wsparcie techniczne jest dostępne w celu zapewnienia klientom obsługi technicznej. Klienci mogą się kontaktować z działem Wsparcia technicznego w celu uzyskania pomocy dotyczącej korzystania z IBM Corp. produktów lub pomocy w instalacji dla jednego z obsługiwanych środowisk sprzętowych. Aby skontaktować się z działem Wsparcia technicznego, wejdź na stronę internetową IBM Corp. pod adresem <http://www.ibm.com/support>. W przypadku prośby o pomoc, należy przygotować swoje dane identyfikacyjne, dane swojej organizacji, a także dane dotyczące usług wsparcia.



---

## Rozdział 1. O programie IBM SPSS Modeler

IBM SPSS Modeler to zestaw narzędzi do eksploracji danych. Produkt umożliwia szybkie opracowywanie modeli predykcyjnych przy wykorzystaniu wiedzy specjalistycznej i stosowanie tych modeli w procesach biznesowych, jako wsparcia przy podejmowaniu decyzji. Rozwiązania zawarte w oprogramowaniu IBM SPSS Modeler zapewniają możliwość wykorzystywania branżowego modelu CRISP-DM i pozwalają na obsługę całego procesu eksploracji danych: od pozyskiwania danych do uzyskiwania lepszych wyników biznesowych.

Oprogramowanie IBM SPSS Modeler umożliwia korzystanie z wielu metod modelowania opartych na sztucznej inteligencji, uczeniu maszynowym i statystykach. Metody dostępne na palecie Modelowanie pozwalają na ekstrakowanie nowych informacji z danych i tworzenie modeli predykcyjnych. Każda metoda ma określone mocne strony i jest dostosowana do rozwiązywania określonych problemów.

Oprogramowanie SPSS Modeler można zakupić jako produkt samodzielny lub jako program kliencki używany wraz z oprogramowaniem SPSS Modeler Server. Dostępnych jest wiele opcji dodatkowych, które przedstawiono w kolejnych rozdziałach. Aby uzyskać więcej informacji, patrz <https://www.ibm.com/analytics/us/en/technology/spss/>.

---

### Produkty IBM SPSS Modeler

Rodzina produktów IBM SPSS Modeler i towarzyszącego im oprogramowania składa się z elementów przedstawionych poniżej.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (jest częścią produktu IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services

### IBM SPSS Modeler

Oprogramowanie SPSS Modeler to w pełni funkcjonalna wersja produktu instalowana i uruchamiana na komputerze osobistym. Oprogramowanie SPSS Modeler można uruchomić lokalnie jako produkt samodzielny lub korzystać z niego w trybie rozproszonym wraz z serwerem IBM SPSS Modeler Server. Tego typu rozwiązanie zapewnia zwiększenie wydajności obsługi dużych zbiorów danych.

Dzięki oprogramowaniu SPSS Modeler można szybko tworzyć dokładne modele predykcyjne, stosując intuicyjne metody niewymagające umiejętności programowania. Unikatowy interfejs graficzny pozwala na wizualizowanie procedur eksploracji danych. Zaawansowane metody opracowywania analiz dostępne w programie umożliwiają określanie wcześniej niezauważalnych wzorców i trendów zawartych w danych. Użytkownik może modelować wyniki i poznawać czynniki wpływające na ich wartości. W ten sposób można wykorzystywać nowe szanse biznesowe i obniżać ryzyko.

Dostępne są dwie edycje oprogramowania SPSS Modeler: SPSS Modeler Professional oraz SPSS Modeler Premium. Więcej informacji można znaleźć w temacie “Wydania programu IBM SPSS Modeler” na stronie 2.

### IBM SPSS Modeler Server

Oprogramowanie SPSS Modeler działa w oparciu o architekturę klient-serwer, w której żądania wymagające zaangażowania dużych zasobów kierowane są do zaawansowanego oprogramowania serwerowego. Takie rozwiązanie umożliwia bardziej wydajną obsługę dużych zbiorów danych.

SPSS Modeler Server to produkt wymagający dodatkowej licencji, działający stale na serwerze w trybie analizy rozproszonej. Współpracuje on z co najmniej jedną instalacją oprogramowania IBM SPSS Modeler. W ten sposób oprogramowanie SPSS Modeler Server poprawia wydajność podczas obsługi dużych zbiorów danych, ponieważ operacje wymagające dużej mocy obliczeniowej można wykonywać na serwerze bez potrzeby pobierania danych na komputer kliencki. Oprogramowanie IBM SPSS Modeler Server optymalizuje również obsługę SQL i funkcje modelowania wewnątrz bazy danych, co dodatkowo zwiększa wydajność działania i sprzyja automatyzacji pracy.

## **IBM SPSS Modeler Administration Console**

Oprogramowanie Modeler Administration Console to graficzny interfejs użytkownika służący do obsługi wielu opcji konfiguracji SPSS Modeler Server, które można dostosować również za pomocą pliku opcji. Konsola udostępniona w aplikacji IBM SPSS Deployment Manager pozwala na monitorowanie i konfigurowanie instalacji SPSS Modeler Server. Konsola jest dostępna bez dodatkowych opłat dla aktualnych użytkowników SPSS Modeler Server. Aplikację można zainstalować tylko na komputerach z systemem Windows, jednak administrować można serwerem zainstalowanym na dowolnej obsługiwanej platformie.

## **IBM SPSS Modeler Batch**

Eksploatacja danych jest zazwyczaj procesem interaktywnym, jednak oprogramowanie SPSS Modeler można też uruchomić z poziomu wiersza komend i zrezygnować z używania graficznego interfejsu użytkownika. Niekiedy użytkownik wykonuje długotrwałe lub powtarzalne zadania, które mogą być realizowane bez nadzoru. Oprogramowanie SPSS Modeler Batch to specjalna wersja produktu pozwalająca na wykonywanie wszystkich funkcji analitycznych SPSS Modeler bez potrzeby używania standardowego interfejsu użytkownika. Oprogramowanie SPSS Modeler Server jest wymagane do korzystania z aplikacji SPSS Modeler Batch.

## **IBM SPSS Modeler Solution Publisher**

SPSS Modeler Solution Publisher umożliwia tworzenie spakowanych wersji strumieni programu SPSS Modeler, które można uruchamiać za pomocą zewnętrznych środowisk wykonawczych lub osadzać w aplikacji zewnętrznej. W ten sposób można publikować i wdrażać pełne strumienie SPSS Modeler w celu używania ich w środowiskach, w których nie zainstalowano programu SPSS Modeler. SPSS Modeler Solution Publisher jest dystrybuowany jako część produktu IBM SPSS Collaboration and Deployment Services - Scoring, który do działania wymaga oddzielnej licencji. Wraz z licencją użytkownik otrzymuje oprogramowanie SPSS Modeler Solution Publisher Runtime umożliwiające uruchamianie opublikowanych strumieni.

Więcej informacji na temat programu SPSS Modeler Solution Publisher znajduje się w dokumentacji produktu IBM SPSS Collaboration and Deployment Services. W Centrum wiedzy IBM SPSS Collaboration and Deployment Services dostępne są sekcje "IBM SPSS Modeler Solution Publisher" oraz "IBM SPSS Analytics Toolkit".

## **IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services**

Dostępnych jest wiele adapterów dla IBM SPSS Collaboration and Deployment Services, które umożliwiają współpracę programów SPSS Modeler i SPSS Modeler Server z repozytorium IBM SPSS Collaboration and Deployment Services. Dzięki temu strumień SPSS Modeler wdrożony w repozytorium można udostępnić wielu użytkownikom lub uzyskać do niego dostęp z poziomu uproszczonej aplikacji klienckiej IBM SPSS Modeler Advantage. Adapter należy zainstalować na systemie hostującym repozytorium.

---

## **Wydania programu IBM SPSS Modeler**

Dostępne są następujące wydania oprogramowania SPSS Modeler.

### **SPSS Modeler Professional**

Oprogramowanie SPSS Modeler Professional zapewnia wszystkie narzędzia wymagane do obsługi większości typów danych ustrukturyzowanych, takich jak np. zachowania i interakcje śledzone w systemach CRM, dane demograficzne, zachowania zakupowe i dane sprzedażowe.



## SPSS Modeler Premium

Oprogramowanie SPSS Modeler Premium wymaga oddzielnej licencji. Dzięki niemu oprogramowanie SPSS Modeler Professional może obsługiwać wyspecjalizowane dane oraz nieustrukturyzowane dane tekstowe. SPSS Modeler Premium zawiera IBM SPSS Modeler Text Analytics:

Program **IBM SPSS Modeler Text Analytics** korzysta z zaawansowanych rozwiązań lingwistycznych oraz przetwarzania języka naturalnego w celu szybkiego przetwarzania różnego rodzaju nieustrukturyzowanych danych tekstowych, wyodrębniania i porządkowania kluczowych pojęć oraz grupowania tych pojęć w kategorie. Wyodrębnione pojęcia i kategorie można łączyć z istniejącymi danymi ustrukturyzowanymi, takimi jak dane demograficzne, a następnie stosować w celu modelowania, korzystając z produktu IBM SPSS Modeler i zawartego w nim pełnego pakietu narzędzi do eksploracji danych, aby w rezultacie takiego połączenia podejmować lepsze decyzje przy zmniejszonej ilości zakłóceń.

## IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription oferuje te same funkcje analiz predykcyjnych, co tradycyjny klient IBM SPSS Modeler. Użytkownicy edycji Subscription mogą regularnie pobierać aktualizacje produktu.

---

## Dokumentacja

Dokumentacja jest dostępna w programie SPSS Modeler z poziomu menu Pomoc. Spowoduje to otwarcie Centrum Wiedzy, które jest powszechnie dostępne poza produktem.

Pełna dokumentacja dla każdego produktu (obejmująca instrukcje instalacji) jest dostępna również w formacie PDF, w osobnym, skompresowanym folderze, w ramach materiałów dotyczących produktu do pobrania. Dokumenty PDF można również pobrać z Internetu pod adresem <http://www.ibm.com/support/docview.wss?uid=swg27046871>.

## Dokumentacja SPSS Modeler Professional

Pakiet dokumentacji produktu SPSS Modeler Professional (bez instrukcji instalacyjnych) zawiera następujące publikacje.

- **IBM SPSS Modeler — podręcznik użytkownika.** Ogólne wprowadzenie do obsługi oprogramowania SPSS Modeler, w tym opisy procedur tworzenia strumieni danych, obsługi braków danych, tworzenia wyrażeń CLEM pracy z projektami i raportami, a także przygotowywania strumieni do wdrożenia w IBM SPSS Collaboration and Deployment Services lub IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler — węzły źródłowe, procesowe i wyników.** Opisy wszystkich węzłów używanych do odczytywania, przetwarzania i tworzenia wynikowych postaci danych w różnych formatach. Czyli wszystkich węzłów poza węzłami modelowania.
- **IBM SPSS Modeler — Węzły modelowania.** Opisy wszystkich węzłów używanych do tworzenia modeli eksploracji danych. Oprogramowanie IBM SPSS Modeler umożliwia korzystanie z wielu metod modelowania opartych na sztucznej inteligencji, uczeniu maszynowym i statystykach.
- **IBM SPSS Modeler — podręcznik zastosowań.** Przykłady zawarte w niniejszym przewodniku stanowią skrócone informacje związane z konkretnymi metodami i technikami modelowania. Wersja elektroniczna tego podręcznika jest również dostępna z poziomu menu Pomoc. Więcej informacji można znaleźć w temacie “Przykłady zastosowań” na stronie 4.
- **IBM SPSS Modeler — podręcznik tworzenia skryptów w języku Python i automatyzacji.** Informacje na temat automatyzacji działania systemu za pomocą skryptów Python wraz z właściwościami służącymi do obsługi węzłów i strumieni.
- **IBM SPSS Modeler — podręcznik wdrażania.** Informacje na temat uruchamiania strumieni IBM SPSS Modeler przedstawione w postaci krokowych operacji wykonywanych podczas przetwarzania zadań w oprogramowaniu IBM SPSS Deployment Manager.
- **IBM SPSS Modeler CLEF Developer's Guide.** Z oprogramowaniem CLEF można zintegrować inne programy pozwalające na przetwarzanie danych lub obsługę algorytmów modelujących w postaci węzłów w IBM SPSS Modeler.

- **IBM SPSS Modeler — podręcznik eksploracji w bazie danych.** Informacje na temat wydajnego wykorzystywania bazy danych w celu zwiększenia wydajności i zakresu funkcji analitycznych za pomocą algorytmów innych firm.
- **IBM SPSS Modeler Server — podręcznik administracji i wydajności.** Informacje na temat konfiguracji i funkcji administracyjnych w oprogramowaniu IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager — Podręcznik użytkownika.** Informacje dotyczące korzystania z interfejsu użytkownika konsoli administracyjnej zawartej w aplikacji Deployment Manager podczas monitorowania i konfigurowania serwera IBM SPSS Modeler Server.
- **IBM SPSS Modeler — podręcznik CRISP-DM.** Szczegółowy podręcznik metodologii CRISP-DM w kontekście eksploracji danych za pomocą oprogramowania SPSS Modeler.
- **IBM SPSS Modeler Batch — podręcznik użytkownika.** Pełny podręcznik obsługi oprogramowania IBM SPSS Modeler w trybie wsadowym obejmujący szczegółowe informacje na temat pracy w trybie wsadowym i korzystania z argumentów z poziomu wiersza komend. Ten podręcznik jest dostępny tylko w formacie PDF.

## Dokumentacja SPSS Modeler Premium

Pakiet dokumentacji produktu SPSS Modeler Premium (bez instrukcji instalacyjnych) zawiera następujące publikacje.

- **SPSS Modeler Text Analytics — podręcznik użytkownika.** Informacje na temat używania analiz tekstu za pomocą oprogramowania SPSS Modeler, obejmują procedury dotyczące węzłów eksploracji tekstu, interaktywnego pulpitu roboczego, szablonów oraz innych zasobów.

---

## Przykłady zastosowań

Podczas gdy narzędzia do eksploracji danych w programie SPSS Modeler mogą pomóc w rozwiązaniu szeregu problemów biznesowych i organizacyjnych, przykłady aplikacji udostępniają krótkie, ukierunkowane wprowadzenia do konkretnych metod i technik modelowania. Używane tutaj zestawy danych są znacznie mniejsze niż ogromne składnice danych zarządzane przez programy do eksploracji danych, lecz używane koncepcje i metody są skalowalne odpowiednio do potrzeb rzeczywistych aplikacji.

Dostęp do przykładów można uzyskać, klikając opcję **Przykłady aplikacji** w menu Pomoc programu SPSS Modeler.

Pliki danych i przykładowe strumienie są instalowane w folderze Dema, w katalogu instalacyjnym produktu. Aby uzyskać więcej informacji, patrz “Folder Demos”.

**Przykłady modelowania w bazach danych.** Przykłady zamieszczono w publikacji *IBM SPSS Modeler — podręcznik eksploracji w bazie danych*.

**Przykłady skryptów.** Przykłady zamieszczono w publikacji *IBM SPSS Modeler Scripting and Automation Guide*.

---

## Folder Demos

Pliki danych i przykładowe strumienie używane z przykładami do aplikacji są instalowane w folderze Demos wewnątrz katalogu instalacyjnego produktu (na przykład: C:\Program Files\IBM\SPSS\Modeler\Plik > Otwórz strumień.

---

## Monitorowanie wykorzystania licencji

Podczas pracy z produktem SPSS Modeler wykorzystanie licencji jest monitorowane i regularnie rejestrowane. Metryka wykorzystania licencji nosi nazwę *AUTHORIZED\_USER* (użytkownik autoryzowany) lub *CONCURRENT\_USER* (użytkownik pracujący jednocześnie), a typ rejestrowanej metryki zależy od typu licencji na produkt SPSS Modeler, którą posiada użytkownik.

Generowane pliki dzienników mogą być przetwarzane przez program IBM License Metric Tool, z którego uzyskać można raporty o wykorzystaniu licencji.

Pliki dzienników wykorzystania licencji są tworzone w tym samym katalogu, w którym zapisywane są dzienniki klienta SPSS Modeler (domyślnie %ALLUSERSPROFILE%/IBM/SPSS/Modeler/<wersja>/log).



---

## Rozdział 2. IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher to wydajne narzędzie do integracji wyników eksploracji danych w procesach biznesowych mających na celu rozwiązywanie rzeczywistych problemów. Za pomocą narzędzia IBM SPSS Modeler Solution Publisher można utworzyć spakowaną wersję strumieni programu, które można uruchamiać za pomocą zewnętrznych środowisk wykonawczych lub osadzać w aplikacji zewnętrznej. Pozwala to wdrażać strumienie modelowania danych w środowisku produkcyjnym w celu obsługiwanie codziennych procesów biznesowych i udostępniania osobom decyzyjnym wiedzy uzyskanej z eksploracji danych.

Użycie narzędzia IBM SPSS Modeler Solution Publisher oferuje więcej możliwości niż w przypadku zwykłego eksportowania modelu (jako PMML), ponieważ umożliwia opublikowanie i wdrożenie *pełnych strumieni IBM SPSS Modeler*. Oznacza to, że można przeprowadzić przygotowanie danych, jak również wykonać operacje na rekordach i zmiennych, takie jak agregacja danych, wybór rekordów lub wyliczenie nowych zmiennych przed utworzeniem predykcji na podstawie modelu. Następnie można dalej przetwarzać wyniki modelu przed zapisaniem danych — wszystko to przez wykonanie opublikowanego strumienia.

*Uwaga:* Publikowanie jest wykonywane przy użyciu karty Publikuj odpowiedniego węzła eksportu. Więcej informacji można znaleźć w temacie “Publikowanie strumieni” na stronie 8. Strumienie zapisane w wersjach produktu wcześniejszych niż 11.0 używały osobnego węzła narzędzia Publisher. Węzeł jest wciąż obsługiwany w celu zapewnienia kompatybilności wstecznej, ale nie jest już dostępny na palecie węzłów.

### Licencjonowanie produktu IBM SPSS Modeler Solution Publisher

*Uwaga:* Produkt IBM SPSS Modeler Solution Publisher jest dystrybuowany jako część produktu IBM SPSS Collaboration and Deployment Services Scoring Service, dla którego wymaga się oddzielnej licencji.  
<http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Po zakupie licencji dostarczany jest osobny program IBM SPSS Modeler Solution Publisher Runtime, który umożliwia wykonywanie opublikowanych strumieni. Więcej informacji można znaleźć w temacie “Instalowanie środowiska wykonawczego” na stronie 15.

---

## Jak działa oprogramowanie IBM SPSS Modeler Solution Publisher

Wdrażanie rozwiązania za pomocą programu IBM SPSS Modeler Solution Publisher obejmuje dwie fazy: *publikowanie* strumienia oraz *wykonywanie* strumienia.

**Publikowanie.** Podczas pracy z procesem eksploracji bazy danych w końcu pojawi się model, który zapewni dobre rozwiązanie dla problemu biznesowego. W tym momencie można zastosować ten model dla procesu biznesowego. Podczas publikowania strumienia na dysku zapisywany jest szczegółowy opis strumienia (jako plik *obrazu* i plik *parametrów*). Więcej informacji można znaleźć w temacie “Publikowanie strumieni” na stronie 8.

**Wykonywanie.** Po opublikowaniu strumienia można odtworzyć proces zaimplementowany w strumieniu przez wykonanie opublikowanego strumienia. Jest to realizowane przez użycie samodzielnego środowiska wykonawczego IBM SPSS Modeler (*modelerrun.exe*) lub stworzenie aplikacji, która używa biblioteki środowiska wykonawczego IBM SPSS Modeler do wykonywania strumienia. Aby używać strumieni poza produktem IBM SPSS Modeler (używając środowiska wykonawczego lub aplikacji użytkownika), należy najpierw zainstalować program IBM SPSS Modeler Solution Publisher Runtime. Więcej informacji można znaleźć w temacie “Instalowanie środowiska wykonawczego” na stronie 15.

---

## Publikowanie strumieni

Publikowanie strumieni odbywa się bezpośrednio z programu IBM SPSS Modeler, przy użyciu standardowych węzłów eksportu: Baza danych, Plik płaski, Eksport Statistics, Eksport do Data Collection, Plik SAS, Excel i XML. Typ węzła eksportu determinuje format wyników zapisywany za każdym razem, gdy opublikowany strumień jest wykonywany przy użyciu programu IBM SPSS Modeler Solution Publisher Runtime lub aplikacji zewnętrznej. Na przykład, jeśli chcesz zapisać wyniki w bazie danych za każdym razem, gdy uruchamiany jest opublikowany strumień, należy użyć węzła eksportu Baza danych.

### Aby opublikować strumień

1. Otwórz lub zbuduj strumień w normalny sposób i dołącz na końcu węzeł eksportu.
2. Na karcie Publikuj w węźle eksportu określ nazwę podstawową publikowanych plików (to jest nazwę plików, do której będą dodawane różne rozszerzenia: .pim, .par i .xml).
3. Kliknij przycisk **Publikuj**, aby opublikować strumień lub wybierz opcję **Opublikuj strumień**, aby automatycznie opublikować strumień za każdym razem, gdy wykonywany jest węzeł.

**Publikowana nazwa.** Określ nazwę podstawową publikowanych plików obrazu i parametrów.

- **Plik obrazu** (\*.pim) zapewnia wszystkie informacje wymagane do wykonania opublikowanego strumienia przez środowisko wykonawcze dokładnie w taki sam sposób, jak w momencie eksportu. Jeśli masz pewność, że nie zajdzie potrzeba zmiany żadnych ustawień strumienia (takich jak źródło danych wejściowych lub plik danych wyjściowych), możesz wdrożyć tylko plik obrazu.
- **Plik parametrów** (\*.par) zawiera umożliwiający dostosowanie informacje o źródłach danych, plikach wyjściowych i opcjach wykonywania. Jeśli chcesz kontrolować dane wejściowe i wyjściowe strumienia bez ponownej publikacji, potrzebny będzie plik parametrów i plik obrazu.
- **Plik metadanych** (\*.xml) opisuje dane wejściowe i wyjściowe obrazu oraz ich modele danych. Ten plik jest przeznaczony do użycia przez aplikacje, które mają osadzoną bibliotekę środowiska wykonawczego i które muszą znać strukturę danych wejściowych i wyjściowych.

**Uwaga:** Ten plik jest generowany tylko, jeśli zaznaczono opcję **Generuj metadane**.

**Parametry publikacji.** Jeśli jest to wymagane, można uwzględnić parametry strumienia w pliku \*.par. Można zmienić te wartości parametrów strumienia podczas wykonywania obrazu, edytując plik \*.par lub przez środowisko wykonawcze API.

Ta opcja włącza przycisk **Parametry**. Okno dialogowe Publikuj parametry jest wyświetlane po kliknięciu przycisku.

Wybierz parametry, które chcesz uwzględnić w publikowanym obrazie, zaznaczając odpowiednią opcję w kolumnie **Publikuj**.

**Przy wykonaniu strumienia.** Określa, czy strumień jest automatycznie publikowany przy wykonywaniu węzła.

- **Eksportuj dane.** Wykonuje węzeł eksportu w standardowy sposób bez publikowania strumienia. (Węzeł jest wykonywany w programie IBM SPSS Modeler w taki sam sposób, jak gdyby nie było dostępne oprogramowanie IBM SPSS Modeler Solution Publisher). Jeśli wybierzesz tę opcję, strumień nie zostanie opublikowany, o ile nie zrobisz tego samodzielnie, klikając przycisk **Publikuj** w oknie dialogowym węzła eksportu. Można też opublikować bieżący strumień, używając narzędzia Publikuj na pasku narzędzi lub używając skryptu.
- **Opublikuj strumień.** Publikuje strumień do wdrożenia, używając programu IBM SPSS Modeler Solution Publisher. Zaznacz tę opcję, jeśli chcesz automatycznie publikować strumień za każdym razem, gdy jest wykonywany.

**Uwaga:**

- Jeśli planujesz uruchomić opublikowany strumień z użyciem nowych lub aktualizowanych danych, ważne jest, aby zauważyć, że kolejność zmiennych w pliku danych wejściowych musi być taka sama, jak kolejność zmiennych w pliku danych wejściowych węzła źródłowego określonego w opublikowanym strumieniu.

- Podczas publikowania do aplikacji zewnętrznych należy rozważyć filtrowanie zmiennych zewnętrznych lub zmianę nazw zmiennych, aby zapewnić zgodność z wymogami danych wejściowych. Oba rozwiązania można osiągnąć za pomocą węzła Filtruj przed węzłem eksportu.





---

## Rozdział 3. Wykonywanie opublikowanych strumieni

Głównym efektem wykonania opublikowanego strumienia jest zawsze wygenerowanie zestawu danych. Dane mogą być zapisywane w pliku danych lub w bazie danych. Opublikowane strumienie nie mogą generować wykresów, tabel, wygenerowanych modeli i innych elementów wyjściowych bez danych.

Strumienie opublikowane za pomocą programu IBM SPSS Modeler Solution Publisher można wykonać, używając programu IBM SPSS Modeler Solution Publisher Runtime. Program środowiska wykonawczego *modelerrun.exe* jest uruchamiany z wiersza komend z opcjami wskazującymi plik obrazu i (opcjonalnie) plik parametrów. Komenda środowiska wykonawczego jest następująca:

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

gdzie wpisy komendy przedstawiono w poniższej tabeli.

Tabela 1. Wpisy komendy środowiska wykonawczego

| Wpis komendy     | Opis   |
|------------------|--|
| -nobanner        | wstrzymuje początkowy komunikat tekstowy dla środowiska wykonawczego (opcjonalnie).  |
| <parameter-file> | to nazwa pliku parametrów (opcjonalnie)  |
| <options>        | to rozdzielona przecinkami lista par w formie opcja=wartość określających opcje wykonania w środowisku wykonawczym. Opcje opisano poniżej (opcjonalnie). <b>Uwaga:</b> po przecinku nie może występować puste miejsce. |
| <image-file>     | to nazwa pliku obrazu, który zostanie wykonany (wymagane).   |

Dostępne opcje wykonania przedstawiono w poniższej tabeli.

Tabela 2. Prawidłowe opcje środowiska wykonawczego.

| Nazwa                 | Wartość domyślna | Opis   |
|-----------------------|------------------|--|
| max_file_size         | -1               | Maksymalny rozmiar zapisanych plików. Wartość -1 oznacza brak limitu.  |
| max_sql_string_length | 2048             | Maksymalna długość łańcucha zaimportowanego z bazy danych za pomocą SQL. Wartości łańcucha dłuższe niż ta wartość są obcinane po prawej stronie bez ostrzeżenia. Poprawny jest zakres od 1 do 65 535 znaków. |
| memory_usage          | 100              | Mnożnik dynamicznej alokacji pamięci. Zmień tę wartość w górę lub w dół, aby wyregulować całkowite użycie pamięci przez serwer.  |
| temp_directory        | "                | Określa katalog używany dla plików tymczasowych.   |
| request_passwords     | Y                | Kontroluje, czy użytkownik jest monitowany o hasło bazy danych, gdy uruchamia opublikowany strumień, który wymaga dostępu do bazy danych (Y/N).  |

**Uwaga:** Gdy w programie **SPSS Modeler Solution Publisher** wykonywany jest strumień zawierający węzeł optymalizacji CPLEX, domyślnie używana jest wbudowana edycja Community biblioteki CPLEX. W edycji tej

obowiązuje ograniczenie do 1000 zmiennych i 1000 ograniczeń. Jeśli zainstalowana jest pełna edycja programu IBM ILOG CPLEX, w której takie ograniczenie nie obowiązuje, można skorzystać właśnie z niej, wykonując następujące czynności (w zależności od używanej platformy):

- W systemie Windows dodaj ścieżkę biblioteki OPL jako argument wywołania programu modelerrun.exe. Na przykład:

```
-o cplex_opl_lib_path="<ścieżka_CPLEX>\opl\bin\<katalog_Platformy>"
```

Gdzie <ścieżka\_CPLEX> jest katalogiem instalacyjnym produktu CPLEX, na przykład C:\Program Files\IBM\ILOG\CPLEX\_Studio127, a <katalog\_Platformy> jest katalogiem właściwym dla platformy, na przykład x64\_win64.

- W systemie Linux zmodyfikuj plik modelerrun, dodając ścieżkę do biblioteki OPL. Na przykład:

```
CPLEX_OPL_LIB_PATH=<ścieżka_CPLEX>/opl/bin/<katalog_Platformy>
```

Gdzie <ścieżka\_CPLEX> jest katalogiem instalacyjnym oprogramowania CPLEX, np. /root/Libs\_127\_FullEdition/Linux\_x86\_64, a <katalog\_Platformy> jest katalogiem właściwym dla platformy, np. x86-64\_linux.

## Uruchamianie lokalnego strumienia Apache Spark w programie IBM SPSS Modeler Solution Publisher

Uruchamiając lokalny strumień Spark w programie IBM SPSS Modeler Solution Publisher, należy określić następującą ścieżkę modelerrun (gdzie C:/Anaconda2/python.exe jest ścieżką do wykonywalnego pliku Python):

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

---

## Używanie pliku parametrów do dostosowania wykonywania strumienia

Kiedy strumień jest publikowany, plik obrazu zawiera informacje o źródle danych i celu danych wyjściowych, które były wybrane w oryginalnym strumieniu. Często jednak korzystnie jest uruchomić strumień dla danych z innego źródła danych lub przekierować dane wyjściowe do innego miejsca docelowego. Można to zrobić, edytując plik parametrów strumienia.

Jest to szczególnie ważne, jeśli opublikowany strumień będzie wykonywany na innym komputerze, niż ten, na którym został utworzony. Należy aktualizować lokalizacje plików wejściowych i wyjściowych, aby odzwierciedlić strukturę komputera docelowego. Należy pamiętać, że opublikowane strumienie nie obsługują ścieżek względnych, co powoduje, że bardzo ważne jest sprawdzenie poprawności lokalizacji danych w pliku parametrów. (Jeśli określisz ścieżki względne w opcjach węzła Eksportuj podczas publikacji strumienia, program IBM SPSS Modeler przekształci je wcześniej na ścieżki bezwzględne, tworząc pliki obrazu i parametrów dla publikowanego strumienia.)

Możesz również użyć pliku parametrów do ustawienia opcji wykonywania i zmienić wartości parametrów strumienia.

Wartości parametrów są określane w pliku parametrów jako pary w formie <nazwa.atrybut>=<wartość>. Dla niektórych parametrów nazwa parametru zawiera cyfrę identyfikatora, aby wyróżnić węzły tego samego typu w strumieniu. Na przykład w strumieniu, w którym zarówno dane wejściowe, jak i wyjściowe dotyczą połączenia bazy danych, parametry opisujące połączenie wejściowe mogą mieć nazwy dbconn0.datasource, dbconn0.user itp., a parametry powiązane z połączeniem wyjściowym mogą mieć nazwy dbconn1.datasource, dbconn1.user, itp. Wartości są zawsze podawane w cudzysłowach. Plik parametrów zawiera następujące parametry, z których wszystkie można zmienić, aby dostosować wykonywanie strumienia.

Tabela 3. Parametry wykonywania.

| Parametr           | Cel  |
|--------------------|--|
| dbconnx.datasource | Nazwa źródła danych (DSN).                                 |
| dbconnx.user       | Nazwa użytkownika dla bazy danych z ograniczonym dostępem. |
| dbconnx.password   | Hasło do bazy danych z ograniczonym dostępem.              |

Tabela 3. Parametry wykonywania (kontynuacja).

| Parametr                     | Cel   |
|------------------------------|---|
| dbconnx.epassword            | Zakodowane hasło do bazy danych z ograniczonym dostępem. Aby wygenerować zakodowane hasło, wybierz pozycję <b>Kodowanie hasła</b> z menu Narzędzia interfejsu użytkownika programu IBM SPSS Modeler. Skopiuj i wklej zakodowane hasło zgodnie z potrzebami. |
| dbconnx.password_required    | Flaga wskazująca, czy zapytać o hasło dla tego połączenia z bazą danych. (Wartość to Y lub N.) Automatycznie ustaw na N, jeśli nie określono hasła w momencie publikacji.   |
| dbconnx.catalog              | Katalog powiązany z połączeniem z bazą danych.  |
| dbtablex.name                | Nazwa tabeli podstawowej. <i>Uwaga:</i> kolejność zmiennych w nowej tabeli musi być taka sama, jak kolejność zmiennych w oryginalnie opublikowanym strumieniu.  |
| dbtablex.schema              | Nazwa schematu.   |
| dbtablex.catalog             | Katalog powiązany z połączeniem z bieżącą tabelą.   |
| filex.name                   | Nazwa pliku dla pliku danych. <i>Uwaga:</i> kolejność zmiennych w nowym pliku wejściowym musi być taka sama, jak kolejność zmiennych w oryginalnie opublikowanym strumieniu.  |
| filex.path                   | Lokalizacja pliku (nazwa katalogu).   |
| filex.field_names_included   | Dla plików w formacie tekstowym flaga wskazująca, czy nazwy zmiennych są uwzględnione jako pierwszy wiersz pliku. (Wartość to Y lub N.)   |
| filex.field_separator        | Dla plików w formacie tekstowym znaki używane do rozdzielenia wartości zmiennych.   |
| filex.decimal_separator      | Dla plików w formacie tekstowym znaki używane jako znak dziesiętny.   |
| options.angle_in_radians     | Flaga wskazująca, czy używane są radiany jako jednostka miary w wyrażeniach trygonometrycznych CLEM. (Wartość to Y lub N.)  |
| options.date_2digit_baseline | Definiuje wiek dla dat z rokiem zapisanym za pomocą dwóch cyfr.   |
| options.date_baseline        | Rok bazowy (zawsze 1 stycznia) używany przez funkcje daty CLEM obsługujące pojedynczą datę.   |
| options.time_rollover        | Flaga wskazująca, czy ujemne różnice czasu dotyczą przeszłości.   |
| options.decimal_separator    | Domyślny znak separatora dziesiętnego w plikach tekstowych.   |
| options.time_format          | Format czasu używany, gdy łańcuchy są interpretowane jako czas przez funkcje czasu CLEM.  |
| options.date_format          | Format daty używany, gdy łańcuchy są interpretowane jako daty przez funkcje dat CLEM.   |
| options.timestamp_format     | Format używany podczas odczytu zmiennych znaczników czasu jako łańcuchów ze źródeł danych ODBC.   |
| paramx.value                 | Wartość parametru strumienia.<br><i>Uwaga:</i> nazwy parametru, opis i typ składowania są zawarte w celach referencyjnych, ale nie można ich zmienić. Wartość musi być prawidłowa dla typu.   |

## Osadzanie produktu IBM SPSS Modeler Solution Publisher w aplikacjach

Oprócz samodzielnego silnika środowiska wykonawczego produkt IBM SPSS Modeler Solution Publisher udostępnia bibliotekę oprogramowania środowiska wykonawczego (CLEMRTL), która pozwala innym programom na kontrolowanie wykonywania strumieni IBM SPSS Modeler Solution Publisher. Procedury CLEMRTL można wywołać w programach klientów napisanych w języku C i C++. Aby użyć biblioteki CLEMRTL, należy uwzględnić plik

nagłówka *clmrtl.h* (dostępny w folderze `..\installation\clmrtl\include`) i powiązać odpowiedni plik biblioteki dla platformy programistycznej w aplikacji, gdy jest budowana. Plik *clmrtl.dll* jest dostępny z folderu `..\installation\bin`.

Każdy plik źródłowy, który odnosi się do procedur biblioteki, musi zawierać nagłówek *clmrtl.h*. Ten plik nagłówka zawiera prototypy ANSI C dla procedur biblioteki i definiuje przydatne makra. Plik nie wymaga załączenia żadnych innych nagłówków oprócz tych, których wymaga program. Aby uchronić przed konfliktami nazw, wszystkie typy biblioteki i nazwy funkcji zaczynają się od ciągu `clmrtl_`, a wszystkie nazwy makr są poprzedzane ciągiem `CLEMRTL_`.

Biblioteka CLEMRTL generuje raporty zawierające przydatne informacje, które powinny być przekazane z powrotem do aplikacji. Biblioteka CLEMRTL udostępnia trzy mechanizmy obsługi takich komunikatów:

- Komunikaty można zapisać w pliku dziennika określonym za pomocą funkcji `clmrtl_setLogFile()`.
- Aplikacja może pobrać szczegóły ostatniego błędu, używając funkcji `clmrtl_getErrorDetail()`.
- Aplikacja może udostępnić własną procedurę obsługi raportów, używając funkcji `clmrtl_setReportHandler()`.

System środowiska wykonawczego posiada zlokalizowany katalog komunikatów dla raportów. Każdy raport przekazany do aplikacji będzie zawierał zlokalizowany łańcuch komunikatu. Raporty zawierają również kody raportów, aby aplikacja mogła zinterpretować i zaprezentować komunikat w inny sposób. W celu uzyskania szczegółowych (zlokalizowanych) komunikatów aplikacja potrzebuje odpowiedniego pliku *messages.cfg* w katalogu *config*. Różne wersje pliku *messages.cfg* dla różnych regionów można znaleźć w katalogu `<installdir>/config/<region>`, gdzie `<installdir>` to katalog, w którym zainstalowany jest produkt IBM SPSS Modeler Solution Publisher, a `<region>` to wybrany region. Wybierz odpowiednią wersję pliku *messages.cfg* i dodaj ją do podkatalogu *config* w katalogu projektu.

System środowiska wykonawczego nie potrzebuje specjalnego środowiska lub ustawień rejestru, aby działał prawidłowo. Biblioteki zależne muszą być rozłożone za pomocą aplikacji powiązanej z biblioteką CLEMRTL. Te pliki są zawarte w katalogu `<installdir>/bin`.

## Wymagania

Poniższa tabela przedstawia platformy i kompilatory przetestowane z biblioteką programowania środowiska wykonawczego.

Tabela 4. Platformy i kompilatory.

| Platforma                | Sugerowane kompilatory                       |
|--------------------------|--|
| AIX                      | IBM XL C/C++ Enterprise Edition V8.0 for AIX |
| Linux — wersja 64-bitowa | gcc 3.2.3, Intel C++ compiler 9.1.043        |
| Windows                  | Visual Studio 2008 R2                        |

## Przykładowa aplikacja IBM SPSS Modeler Solution Publisher

Przykładowa aplikacja *dlltest.c* znajduje się w folderze `\clmrtl\demo` w katalogu instalacyjnym IBM SPSS Modeler Solution Publisher. Przykładowa aplikacja pomoże rozpocząć prace z osadzaniem biblioteki środowiska wykonawczego we własnych aplikacjach.

Informacje używane przez aplikację przykładową są zawarte w folderze `\clmrtl\demo`. Instrukcje można znaleźć w pliku *README.txt*. Kompilowanie aplikacji różni się w zależności od platformy, więc załączono przykładowe pliki *makefile*.

---

## Rozdział 4. Instalowanie produktu IBM SPSS Modeler Solution Publisher Runtime

---

### Instalowanie środowiska wykonawczego

Zainstalowanie programu IBM SPSS Modeler Solution Publisher Runtime pozwala na wykonywanie opublikowanych strumieni. Opublikowane strumienie są niezależne od systemu operacyjnego, można więc wykonywać dowolne opublikowane strumienie na dowolnej platformie obsługiwanej przez środowisko wykonawcze.

Funkcje pozwalające na publikowanie strumieni z programu IBM SPSS Modeler są instalowane razem z produktem IBM SPSS Modeler. Należy pamiętać, że opublikowane pliki muszą być eksportowane ponownie przy każdej wersji programu IBM SPSS Modeler. Na przykład pliki eksportowane za pomocą wcześniejszej wersji nie będą działały w bieżącej wersji programu IBM SPSS Modeler Solution Publisher Runtime.

### Instalowanie w systemie Windows

Poniższa sekcja opisuje procedurę instalowania produktu SPSS Modeler Solution Publisher Runtime w systemie operacyjnym Windows.

1. Uruchom instalację programu IBM SPSS Modeler Solution Publisher.
2. W menu instalowania wybierz opcję **Zainstaluj IBM SPSS Modeler Solution Publisher Runtime** lub **Zainstaluj IBM SPSS Modeler Solution Publisher Runtime for Win64** w zależności od systemu, w którym przeprowadzana jest instalacja.
3. W kreatorze instalacji kliknij przycisk **Dalej**, aby rozpocząć.
4. Następnie postępuj zgodnie z instrukcjami wyświetlanymi na ekranie. Aby kontynuować, kliknij przycisk **Dalej**.
5. Po określeniu wszystkich opcji można przystąpić do instalowania. Kliknij przycisk **Zainstaluj**, aby rozpocząć przesyłanie plików.
6. Po zainstalowaniu wszystkich plików kliknij przycisk **Zakończ**.

### Instalowanie w systemie UNIX

Poniższe sekcje opisują dodatkowe wymagania systemowe oraz procedurę instalowania produktu IBM SPSS Modeler Solution Publisher Runtime w systemie operacyjnym UNIX.

#### Dodatkowe wymagania dla systemu UNIX

Upewnij się, że limity jądra w systemie są wystarczające i umożliwiają działanie serwera IBM SPSS Modeler Server. Wartości ulimit dotyczące danych, pamięci plików i procesów są szczególnie istotne i powinny być ustawione na nieograniczone w środowisku IBM SPSS Modeler Server. W tym celu:

1. Dodaj następujące komendy do pliku *modelersrv.sh*:

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

Dodatkowo należy ustawić limit stosu na maksymalny dopuszczalny systemowo (ulimit -s XXXX), na przykład:

```
ulimit -s 64000
```

2. Ponownie uruchom system IBM SPSS Modeler Server.

Potrzebne jest również narzędzie kompresji plików *gzip* oraz *GNU cpio* zainstalowane na ścieżce, aby instalator mógł rozpakować pliki instalacyjne. Dodatkowo na komputerze, na którym działa produkt SPSS Modeler Server należy określić ustawienia lokalne jako EN\_US.UTF-8.

## Procedura instalowania w systemie UNIX

Te instrukcje instalowania dotyczą produktu IBM SPSS Modeler Solution Publisher Runtime dla systemu UNIX. Potrzebne są uprawnienia do odczytu i zapisu w docelowym katalogu instalowania, więc należy zalogować się na koncie z wystarczającymi uprawnieniami.

**Uwaga:** Użytkownik musi być zalogowany jako *root*, aby przeprowadzić instalację.

1. W pakiecie instalacyjnym otwórz katalog *modelrun*.
2. Zmień na katalog odpowiedniej platformy.
3. Uruchom skrypt instalacyjny *.bin*. Na przykład:  
`./modelersolutionpublisherhpia64.bin -i console`
4. Wyświetli się wstęp. Naciśnij klawisz Enter, aby kontynuować.
5. Wyświetlą się informacje na temat licencji. Przeczytaj licencję, wpisz 1, aby ją zatwierdzić, i naciśnij klawisz Enter, aby kontynuować.
6. Wyświetli się lista dostępnych języków. Wprowadź liczbę dla wersji językowej, którą chcesz zainstalować, i naciśnij klawisz Enter, aby kontynuować.
7. Wyświetli się monit o wpisanie katalogu instalacyjnego. Aby użyć katalogu domyślnego */usr/IBM/SPSS/ModelerSolutionPublisher<nn>* (gdzie *<nn>* to numer wersji), naciśnij klawisz Enter. Środowisko wykonawcze zostanie zainstalowane w określonym katalogu.
8. Wyświetli się monit o potwierdzenie katalogu instalacyjnego. Jeśli dane są prawidłowe, wpisz y i naciśnij klawisz Enter.
9. Wyświetli się wstępne podsumowanie instalacji w celu zatwierdzenia wpisanych do tej pory danych. Naciśnij klawisz Enter, aby kontynuować.
10. Wyświetli się komunikat mówiący o gotowości procedury instalacyjnej. Naciśnij klawisz Enter, aby kontynuować.
11. Podczas trwania procedury instalacyjnej wyświetlał się będzie pasek postępu. Po skończeniu instalacji naciśnij Enter, aby wyjść z instalatora.

## Konfiguracja produktu ODBC w systemie UNIX

Domyślnie narzędzie DataDirect Driver Manager nie jest skonfigurowane dla programu IBM SPSS Modeler Solution Publisher Runtime do używania ODBC w systemach UNIX. Aby skonfigurować system UNIX do ładowania narzędzia DataDirect Driver Manager, wprowadź następujące komendy (gdzie *sp\_install\_dir* to katalog instalacyjny programu Solution Publisher Runtime):

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

## Rozwiązywanie problemów i instalacja

### Nieprawidłowy podpis cyfrowy podczas instalacji

Produkty IBM SPSS Modeler używają certyfikatów wystawionych przez IBM do podpisów cyfrowych. W niektórych okolicznościach można zobaczyć następujący błąd przy próbie zainstalowania produktów SPSS Modeler:

Błąd 1330. Nie można zainstalować wymaganego pliku, ponieważ nazwa pliku CAB ma nieprawidłowy podpis cyfrowy...

### Wszyscy użytkownicy systemów Windows

Ten komunikat jest wyświetlany podczas próby zainstalowania produktów SPSS Modeler na komputerze, który nie ma połączenia internetowego i nie ma zainstalowanych prawidłowych certyfikatów. Użyj poniższej procedury, aby naprawić ten problem.

1. Kliknij przycisk **OK**, aby zatwierdzić komunikat.
2. Kliknij przycisk **Anuluj**, aby wyjść z instalatora.

3. Jeśli komputer, na którym chcesz przeprowadzić instalację, nie ma połączenia internetowego, wykonaj następujący krok na komputerze podłączonym do Internetu i skopiuj plik .cer na komputer, na którym chcesz przeprowadzić instalację.
4. Przejdź na stronę <https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&id=SO5624> i postępuj zgodnie z instrukcjami, aby pobrać **certyfikat główny VeriSign Class 3 Primary Certification Authority - G5**. Zapisz go jako plik .cer.
5. Dwukrotnie kliknij plik .cer.
6. Na karcie Ogólne kliknij opcję **Zainstaluj certyfikat**.
7. Postępuj zgodnie z instrukcjami wyświetlanymi w narzędziu Kreator importu certyfikatów, używając domyślnych opcji, i na końcu kliknij przycisk **Zakończ**.
8. Spróbuj ponowić próbę instalowania.

## Uruchamianie środowiska wykonawczego

Po zainstalowaniu środowiska wykonawczego można go używać do wykonywania strumieni, które były opublikowane w programie IBM SPSS Modeler za pomocą dowolnych węzłów eksportu. Więcej informacji można znaleźć w temacie “Publikowanie strumieni” na stronie 8. Uruchom środowisko wykonawcze z wiersza komend z opcjami wskazującymi nazwę opublikowanego pliku obrazu oraz opcjonalny plik parametrów do wykonania.

Aby uruchomić środowisko wykonawcze, wpisz następujący wiersz w wierszu komend:

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

gdzie wpisy komendy przedstawiono w poniższej tabeli.

Tabela 5. Wpisy komendy środowiska wykonawczego

| Wpis komendy     | Opis   |
|------------------|--|
| <parameter-file> | to nazwa opublikowanego pliku parametrów (opcjonalnie).  |
| <options>        | to rozdzielona przecinkami lista par w formie <i>opcja=wartość</i> określających opcje wykonania w środowisku wykonawczym. <b>Uwaga:</b> po przecinku nie może występować puste miejsce. |
| <image file>     | to nazwa opublikowanego pliku obrazu (*.pim), który zostanie wykonany.   |

*Uwaga:* Podczas używania języków wielobajtowych, takich jak chiński lub japoński, w środowisku UNIX należy określić używane kody języka. W poniższym przykładzie przedstawiono kody języka dla uproszczonego chińskiego:

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

gdzie -p to lokalizacja plików pim i par.

Więcej informacji można znaleźć w Rozdział 3, “Wykonywanie opublikowanych strumieni”, na stronie 11.

## Zmiana położenia katalogu tymczasowego

Wykonanie niektórych operacji w oprogramowaniu IBM SPSS Modeler Solution Publisher Runtime może wymagać utworzenia plików tymczasowych. Domyślnie podczas obsługi plików tymczasowych środowisko wykonawcze korzysta z katalogu plików tymczasowych w systemie operacyjnym. Wykonując poniższe czynności, można zmienić położenie katalogu tymczasowego.

*Uwaga:* Bardzo ważne jest użycie prawidłowej składni, zachowując wyjątkową ostrożność przy odstępach i ścieżce do katalogu.

Założmy, że produkt IBM SPSS Modeler Solution Publisher Runtime jest używany z następującymi opublikowanymi plikami: *webtest.pim* i *webtest.par*. W takiej sytuacji celowe będzie zmodyfikowanie położenia plików tymczasowych, tak aby były zapisywane w katalogu *C:\published\temp*. Aby zmienić położenie katalogu plików tymczasowych używanego przez pliki wykonywalne:

1. W katalogu, gdzie zapisane są pliki *.pim* and *.par*, uruchom komendę:

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

Spowoduje to, że środowisko wykonawcze będzie korzystało z katalogu "*C:\published\temp*" jako katalogu tymczasowego tylko dla tego konkretnego procesu.

---

## Deinstalacja produktu IBM SPSS Modeler Solution Publisher Runtime

Poniższa sekcja opisuje procedurę deinstalowania produktu IBM SPSS Modeler Solution Publisher Runtime.

### Procedura deinstalacji w systemie Windows

1. W Panelu sterowania systemu Windows uruchom opcję **Dodaj lub usuń programy**.
2. Z listy wybierz pozycję **IBM SPSS Modeler Solution Publisher <nn>**.
3. Kliknij przycisk **Usuń**.
4. Następnie postępuj zgodnie z instrukcjami wyświetlanymi na ekranie, aby usunąć środowisko wykonawcze.

*Uwaga:* Użytkownik musi być zalogowany na komputerze jako administrator, aby usuwać programy.

### Procedura deinstalacji w systemie UNIX

Usuń katalog instalacyjny łącznie z wszystkimi zainstalowanymi plikami.



---

## Rozdział 5. IBM SPSS API EPM (osadzone modelowanie predykcyjne)

Interfejs API EPM (osadzonego modelowania predykcyjnego) IBM SPSS definiuje prosty interfejs Java pozwalający zintegrować zaprojektowane artefakty IBM SPSS Modeler we własnych aplikacjach przez osadzenie. Pełna dokumentacja w formacie Javadoc jest dołączona do produktu IBM SPSS Modeler Solution Publisher dla programistów, którzy używają biblioteki.

Uwaga dotycząca następujących ograniczeń przy użyciu interfejsu API EPM (osadzonego modelowania predykcyjnego):

- **Wymagane są unikalne etykiety węzłów źródłowych i węzłów końcowych.** Interfejs API EPM używa etykiet węzłów. Mimo że etykiety zawsze znajdują się w kontekście węzłów źródłowych lub węzłów końcowych, mogą być dwuznaczne.
- **Superwęzły nie są obsługiwane jako węzły początkowe lub końcowe.** Aby zapewnić wspólne źródło dla superwęzłów we wszystkich operacjach, nie są one obsługiwane jako węzły początkowe i końcowe.
- **Ograniczenia nazwy zmiennych wejściowych.** W nazwach zmiennych nie mogą być używane niektóre znaki.

Zobacz poniższe informacje, aby zapoznać się z omówieniem funkcjonalności API. Aby uzyskać więcej informacji, zobacz dokument Javadoc dołączony do produktu.

### Osadzone ocenianie modelu predykcyjnego

**Użycie:** używane przez aplikację do wygenerowania analizy predykcyjnej na żądanie, gdy dostarczone zostaną nowe dane. Aplikacja musi obsługiwać modele odświeżane. W tym trybie oceniania wszystkie węzły źródłowe, jak również węzeł końcowy gałęzi oceniania, są zastępowane przez dane wejściowe i wyjściowe aplikacji.

**Działanie:** przygotowuje gałąź oceniania opracowaną jednorazowo w strumieniu IBM SPSS Modeler dla powtarzanych wywołań, co szybko i wydajnie generuje analizę predykcyjną z danych wejściowych zarządzanych przez aplikację.

### Odświeżanie modelu predykcyjnego

**Użycie:** przeprowadzenie wstępnej nauki algorytmów modelu w projekcie strumienia za pomocą danych historycznych. Dodatkowo okresowe „odświeżanie” skonfigurowanych algorytmów modelu predykcyjnego za pomocą najnowszych danych historycznych.

**Działanie:** uruchamia wszystkie węzły modelu predykcyjnego do utworzenia w strumieniu IBM SPSS Modeler, co uczy algorytm modelu predykcyjnego zgodnie z konfiguracją przy użyciu technik eksploracji danych. Aktualizuje modele użytkowe modelu do zastosowania powiązane z każdym węzłem modelu do utworzenia. Do wdrożenia pewnego stopnia dynamiki można użyć parametrów strumienia i właściwości węzłów źródłowych lub węzłów końcowych.

### Ocena modelu predykcyjnego

**Użycie:** używane przez aplikację do przekazania dokładności, ufności lub innej oceny jakościowej przeszkolonego modelu predykcyjnego zgodnie z planem w strumieniu IBM SPSS Modeler. Te informacje są zazwyczaj używane do kontroli promowania odświeżonego niedawno modelu predykcyjnego w ocenie w aplikacji. Podstawowym powodem, dla którego Odświeżenie modelu zwraca nowy obiekt PredictiveModel, jest umożliwienie porównania danych oceny dla niedawno odświeżonego modelu z oryginalnym modelem.

**Działanie:** uruchamia wszystkie węzły do utworzenia w dokumencie (przede wszystkim nieinteraktywne węzły wyjściowe w paletach Wynik i Wykres programu IBM SPSS Modeler) w strumieniu IBM SPSS Modeler, zwracając

wynikowe dokumenty. Do wdrożenia pewnego stopnia dynamiki można użyć parametrów strumienia i właściwości węzłów źródłowych lub węzłów końcowych.

## Wykonywanie strumienia modelu predykcyjnego

**Użycie:** wszystkie przypadki, których nie można zakwalifikować jako odświeżanie, ocenianie lub dwa tryby oceny, muszą być obsługiwane przy pomocy wykonywania strumienia.

**Działanie:** wykonanie jest kontrolowane przez opcje strumienia IBM SPSS Modeler. Albo w trybie wszystkich węzłów końcowych podobnym do kliknięcia przycisku dużej zielonej strzałki w kliencie IBM SPSS Modeler, albo jako kod w skrypcie na poziomie strumienia. Do wdrożenia pewnego stopnia dynamiki można użyć parametrów strumienia i właściwości węzłów źródłowych lub węzłów końcowych.

## Dane zewnętrzne oceny modelu predykcyjnego

**Użycie:** zazwyczaj określany jako proces wsadowy, ponieważ przetwarzana jest cała partia danych wejściowych.

**Działanie:** uruchamianie gałęzi oceniania zgodnie z planem. Dane wejściowe z węzłów skonfigurowanych źródeł danych i wygenerowanych analiz predykcyjnych są utrwalane w formie kontrolowanej przez węzeł końcowy. Do wdrożenia pewnego stopnia dynamiki można użyć parametrów strumienia i właściwości węzłów źródłowych lub węzłów końcowych.

## Parametry strumienia modelu predykcyjnego oraz specyfikacja właściwości węzłów źródłowych i końcowych

**Użycie:** jednym z częstych sposobów jest zdefiniowanie parametrów, do których odwołują się wyrażenia w gałęziach strumienia. Innym punktem kontroli jest modyfikacja konkretnych właściwości klucza węzła źródłowego lub końcowego w celu zmienienia używanej ścieżki pliku itp.

**Działanie:** pewne aspekty wykonywania strumienia IBM SPSS Modeler można zmodyfikować dynamicznie przez aplikację.

**Uwaga:** Zmiany parametrów muszą być dokonane przed utworzeniem elementu PredictiveScorer (zobacz powyższy opis osadzonej oceny), a właściwości węzłów źródłowych i końcowych nie mają znaczenia dla tego obiektu.

---

## Rozdział 6. Informacje o bibliotece środowiska wykonawczego API IBM SPSS Modeler Solution Publisher

---

### Wstęp

Interfejs API CLEMRTL pozwala na kontrolowanie wykonania opublikowanych strumieni z aplikacji. Ta sekcja opisuje dostępne funkcje API.

Poniżej przedstawiono ogólne problemy przy używaniu interfejsu API:

- Interfejs API posiada połączenia C w celu zapewnienia maksymalnej kompatybilności, ale biblioteka posiada zależności C++. Na niektórych platformach może to oznaczać, że interfejsu API można używać tylko z konsolidatorem C++.
- Funkcję `clemrtl_initialise_ext()` należy wywołać przed użyciem innych funkcji w bibliotece.
- Typ `clemrtl_image_handle` jest używany jako identyfikator obrazu.
- Każda funkcja zwraca kod wskaźnika statusu. Wartości wskaźnika statusu są przedstawione w poniższej tabeli.

Tabela 6. Wartości wskaźnika statusu

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie  |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji  |
| CLEMRTL_ERROR | Niepowodzenie z dodatkowymi informacjami o błędzie dostępnymi przez funkcję <code>clemrtl_getErrorDetail()</code> |

### Przegląd procesu API

Ogólna struktura aplikacji używającej interfejsu API jest następująca:

1. Inicjuj bibliotekę za pomocą funkcji `clemrtl_initialise_ext()`.
2. Sprawdź i zmień opcje wykonywania za pomocą funkcji `clemrtl_getOption()` i `clemrtl_setOption()`.
3. Otwórz obraz, używając funkcji `clemrtl_openImage()` i otrzymaj uchwyt obrazu.
4. Sprawdź i zmień parametry obrazu, używając funkcji `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` i `clemrtl_setParameter()`.
5. Wykonaj obraz, używając funkcji `clemrtl_execute()`. Jeśli ten sam obraz ma być wykonany wiele razy bez zmiany parametrów, należy najpierw użyć funkcji `clemrtl_prepare()`.
6. Zamknij obraz, używając funkcji `clemrtl_closeImage()`.
7. Aby anulować wykonywanie w toku, użyj funkcji `clemrtl_interrupt()`.
8. Aby pobrać informacje o ostatnim błędzie, użyj funkcji `clemrtl_getErrorDetail()`. Aby otrzymać komunikaty o błędach i inne komunikaty diagnostyczne, gdy zostaną przysłane, użyj funkcji `clemrtl_setReportHandler()`.

Biblioteka jest przeznaczona do użytku wielowątkowego z zastrzeżeniem następujących warunków:

- Funkcje globalne, które nie przyjmują uchwytu obrazu jako argumentu, nie mogą być wywoływane jednocześnie.
- Funkcje lokalne dla obrazu, przyjmujące uchwyt obrazu jako argument, nie mogą być wywoływane jednocześnie, gdy są stosowane dla tego samego uchwytu obrazu, z wyjątkiem funkcji `clemrtl_interrupt()`.
- Funkcje lokalne dla obrazu mogą być wywoływane jednocześnie, gdy są stosowane dla różnych uchwytów obrazów.

Oznacza to, że osobne wątki mogą przygotowywać i wykonywać obrazy jednocześnie, pod warunkiem że używają różnych uchwytów obrazów. Różne uchwytów obrazów mogą dotyczyć tego samego obrazu (funkcja `clemrtl_openImage()` tworzy nowy uchwyt podczas każdego wywołania, nawet dla tego samego obrazu), ale należy

zapewnić, że podczas wykonywania wielu instancji tego samego obrazu zmienione zostały parametry obrazu, aby przekierować dane wyjściowe zgodnie z wymaganiami. Jeśli dwa obrazy wykonywane jednocześnie próbują dokonać zapisu w tym samym pliku wyjściowym lub tabeli bazy danych, wyniki będą nieprzewidywalne.

## Funkcje API

Poniżej przedstawiono pełną listę funkcji ujawnionych w interfejsie API.

### inicjowanie

Ta funkcja jest nieaktualna i jest równoznaczna funkcji

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

Nowe programy powinny wywoływać funkcję *initialise\_ext*.

### initialise\_ext

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

Inicjuje program IBM SPSS Modeler Solution Publisher Runtime. *Ta funkcja musi być wywołana przed innymi funkcjami API.*

Tabela 7. Parametry *initialise\_ext*

| Parametr  | Opis   |
|-----------|--|
| flags     | Kontroluje niektóre z aspektów procesu inicjowania. Wartość jest tworzona jako bitowe OR flag opisanych poniżej.   |
| arg_count | Liczba dodatkowych argumentów inicjowania. Ta długość musi być mniejsza lub równa długości tablicy args.   |
| args      | Dodatkowe argumenty inicjowania. Argumenty są opisywane przez typ:<br>typedef struct _clemrtl_init_arg {<br>const char* name;<br>const char* value;<br>} clemrtl_init_arg; |

Zmienna *name* jest nazwą właściwości konfiguracji, a zmienna *value* jest jej wartością. Rozpoznawane są właściwości konfiguracji przedstawione w poniższej tabeli.

Tabela 8. Właściwości konfiguracji

| Nazwa                  | Wartość  |
|------------------------|--|
| installation_directory | Pełna ścieżka do folderu instalacyjnego programu IBM SPSS Modeler Solution Publisher. Folder instalacyjny znajduje się w folderach bin i config. |

Zwraca jeden z kodów błędów przedstawionych w poniższej tabeli.

Tabela 9. Kody błędów

| Wynik        | Opis  |
|--------------|---|
| CLEMRTL_OK   | Powodzenie.   |
| CLEMRTL_FAIL | Niepowodzenie podczas inicjowania bez dostępnych bardziej szczegółowych informacji. |

Parametr *flags* zapewnia kontrolę nad pewnymi aspektami procesu inicjowania. Wartość 0 (lub CLEMRTL\_INIT\_DEFAULTS) określa następujące działanie domyślne:

- Ustaw domyślną strefę czasową, wywołując funkcję `tzset()`.
- Określ wartość ustawień lokalnych `LC_NUMERIC` na `C`.
- Ustaw nowy globalny uchwyt generujący wyjątek, gdy wyczerpana zostanie pamięć.

To działanie można zmodyfikować, określając kombinację flag przedstawionych w następującej tabeli.

Tabela 10. Flagi

| Flaga                                       | Opis   |
|---|--|
| <code>CLEMRTL_INIT_NO_TZ</code>             | Nie ustawiaj strefy czasowej.                          |
| <code>CLEMRTL_INIT_NO_LOCALE</code>         | Nie zmieniaj ustawień lokalnych.                       |
| <code>CLEMRTL_INIT_NO_NEW_HANDLER</code>    | Nie ustawiaj nowego uchwytu.                           |
| <code>CLEMRTL_INIT_LOCAL_NEW_HANDLER</code> | Zlokalizuj nowy uchwyt w ramach każdego wywołania API. |

Biblioteka wymaga, aby ustawienia lokalne `LC_NUMERIC` miały wartość `C` w celu prawidłowego przekształcania pomiędzy liczbami i łańcuchami (format liczb programu IBM SPSS Modeler używa reguł ustawień lokalnych `C`). Obejmuje to przekształcanie liczb odczytywanych i zapisywanych do plików tekstowych.

- Określ `NO_LOCALE`, jeśli aplikacja zależy od ustawień lokalnych `LC_NUMERIC` i nie wymaga przekształcania pomiędzy liczbami i łańcuchami.
- `NO_NEW_HANDLER` i `LOCAL_NEW_HANDLER` wykluczają się wzajemnie. Biblioteka wymaga, aby element `new` generował wyjątek, gdy wyczerpana zostanie pamięć.
- Określ `NO_NEW_HANDLER`, jeśli aplikacja ustawia własny nowy uchwyt lub jeśli istnieje pewność, że kompilator/system środowiska wykonawczego użytkownika wygeneruje domyślnie wyjątek.

Określ `LOCAL_NEW_HANDLER` tylko jeśli aplikacja wymaga, aby element `new` zwracał pusty wskaźnik, gdy wyczerpana zostanie pamięć. *Uwaga:* Użycie tego rozwiązania w środowisku wielowątkowym nie jest bezpieczne.

## getOption

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

Pobiera wartość opcji wykonania.

Tabela 11. Parametry `getOption`

| Parametr                | Opis                             |
|-------------------------|----------------------------------|
| <code>name</code>       | Nazwa opcji.                     |
| <code>value</code>      | Bufor otrzymujący wartość opcji. |
| <code>value_size</code> | Rozmiar bufora wartości.         |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 12. Kody statusów

| Wynik                      | Opis  |
|----------------------------|---|
| <code>CLEMRTL_OK</code>    | Powodzenie.   |
| <code>CLEMRTL_FAIL</code>  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| <code>CLEMRTL_ERROR</code> | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli nazwa opcji nie jest jedną z nazw opcji wykonywania wymienionych powyżej — na przykład `memory_usage` — lub jeśli bufor wartości nie jest wystarczająco duży, aby otrzymać wartość łącznie ze znakiem kończącym null. Wartości opcji są zawsze łańcuchami, nawet jeśli interpretacją jest wartość liczbowa. Więcej informacji można znaleźć w Rozdział 3, “Wykonywanie opublikowanych strumieni”, na stronie 11.

## setOption

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

Ustawia wartość opcji wykonania.

Tabela 13. Parametry `setOption`

| Parametr | Opis          |
|----------|---------------|
| name     | Nazwa opcji.  |
| value    | Wartość opcji |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 14. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli nazwa opcji nie jest jedną z nazw opcji wykonywania wymienionych powyżej, np. `memory_usage`. Wartości opcji są zawsze łańcuchami, nawet jeśli interpretacją jest wartość liczbowa.

## setLogFile

```
int clemrtl_setLogFile(
    int mode,
    const char* log_file);
```

Przekierowuje komunikaty dziennika z wszystkich otwartych kolejno obrazów.

Tabela 15. Parametry `setLogFile`

| Parametr | Opis  |
|----------|---|
| mode     | Tryb rejestrowania musi być jedną z następujących wartości:<br>CLEMRTL_NULL_LOG — wstrzymaj komunikaty.<br>CLEMRTL_STDERROR_LOG — wyślij komunikaty do standardowego wyjścia błędów.<br>CLEMRTL_FILE_LOG — wyślij komunikaty do pliku o nazwie wskazanej parametrem <code>log_file</code> . |
| log_file | Łańcuch zawierający nazwę pliku.  |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 16. Kody statusów

| Wynik        | Opis  |
|--------------|---|
| CLEMRTL_OK   | Powodzenie.   |
| CLEMRTL_FAIL | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |

Tabela 16. Kody statusów (kontynuacja)

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli tryb jest nie jest jednym z rozpoznawanych trybów lub jeśli zażądano rejestrowania, a nie określono nazwy pliku lub jest ona nieprawidłowa.

## openImage

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

Otwiera opublikowany obraz strumienia.

Tabela 17. Parametry openImage

| Parametr   | Opis                     |
|------------|--------------------------|
| image_file | Nazwa pliku obrazu.      |
| param_file | Nazwa pliku parametrów.  |
| handle     | Otrzymuje uchwyt obrazu. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 18. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Plik parametrów może mieć wartość NULL, jeśli nie jest wymagany osobny plik parametrów. Operacja kończy się niepowodzeniem, jeśli plik obrazu nie jest określony, jeśli nazwa pliku jest nieprawidłowa lub jeśli nie można odczytać zawartości. Jeśli operacja powiedzie się, zwróconego uchwytu obrazu można użyć do zidentyfikowania instancji obrazu w kolejnych wywołaniach API.

## closeImage

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

Zamyka uchwyt obrazu i zwalnia zasoby.

Tabela 19. Parametry closeImage

| Parametr | Opis           |
|----------|----------------|
| handle   | Uchwyt obrazu. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 20. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Ta operacja kończy się niepowodzeniem, jeśli dla obrazu trwa inne wywołanie API, zwłaszcza jeśli obraz jest wciąż wykonywany. Jeśli operacja powiedzie się, obraz zostaje zamknięty i nie można użyć uchwytu w kolejnych wywołaniach API za wyjątkiem kolejnego wywołania `cleMrtl_closeImage()`, które nie odnosi skutku.

## enumerateParameters

```
int cleMrtl_enumerateParameters(
    cleMrtl_image_handle handle,
    cleMrtl_parameter_proc proc,
    void* data);
```

Stosuje procedurę wywołania zwrotnego do każdej nazwy i wartości parametru obrazu.

Tabela 21. Parametry `enumerateParameters`

| Parametr | Opis   |
|----------|--|
| handle   | Uchwyt obrazu.   |
| proc     | Procedura wywołania zwrotnego parametru.                 |
| data     | Dane określone przez użytkownika do wywołania zwrotnego. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 22. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Procedura wywołania zwrotnego ma następujący typ:

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

Tabela 23. Parametry procedury wywołania zwrotnego

| Parametr | Opis  |
|----------|---|
| data     | Dane określone przez użytkownika przekazywane do <code>cleMrtl_enumerateParameters()</code> . |
| name     | Nazwa parametru.  |
| value    | Wartość parametru.  |

Procedura jest stosowana dokładnie raz dla każdego parametru obrazu w kolejności przypadkowej.



## getParameter

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

Pobiera wartość parametru obrazu.

Tabela 24. Parametry *getParameter*

| Parametr   | Opis                                 |
|------------|--------------------------------------|
| handle     | Uchwyt obrazu.                       |
| name       | Nazwa parametru.                     |
| value      | Bufor otrzymujący wartość parametru. |
| value_size | Rozmiar bufora wartości.             |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 25. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli nazwa parametru nie odpowiada nazwie żadnego parametru w obrazie lub jeśli bufor wartości nie jest wystarczająco duży, aby otrzymać wartość parametru łącznie ze znakiem kończącym null. Nazwy parametrów używają opisanego powyżej formatu *nazwa.atrybut*, np. *file0.name*, a wartości parametrów są zawsze łańcuchami, nawet jeśli interpretacją jest wartość liczbowa.

## setParameter

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

Ustawia wartość parametru obrazu.

Tabela 26. Parametry *setParameter*

| Parametr | Opis               |
|----------|--------------------|
| handle   | Uchwyt obrazu.     |
| name     | Nazwa parametru.   |
| value    | Wartość parametru. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 27. Kody statusów

| Wynik      | Opis        |
|------------|-------------|
| CLEMRTL_OK | Powodzenie. |

Tabela 27. Kody statusów (kontynuacja)

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli uchwyt obrazu jest nieprawidłowy lub jeśli nazwa parametru nie odpowiada nazwie żadnego parametru w obrazie. Nazwy parametrów używają opisanego powyżej formatu nazwa.atrybut, np. file0.name, a wartości parametrów są zawsze łańcuchami, nawet jeśli interpretacją jest wartość liczbowa.

## getFieldCount

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

Zwraca liczbę zmiennych w źródle wejściowym lub wyjściowym elemencie docelowym.

Tabela 28. Parametry getFieldCount

| Parametr    | Opis   |
|-------------|--|
| handle      | Uchwyt obrazu.   |
| key         | Nazwa danych wejściowych lub wyjściowych do sprawdzenia w formie używanej w pliku parametrów. Klucz może dotyczyć pliku lub bazy danych. |
| field_count | Otrzymuje liczbę zmiennych.  |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 29. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

## getFieldTypes

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

Zwraca typy zmiennych dla źródła wejściowego lub wyjściowego elementu docelowego.

Tabela 30. Parametry getFieldTypes

| Parametr | Opis   |
|----------|--|
| handle   | Uchwyt obrazu.   |
| key      | Nazwa danych wejściowych lub wyjściowych do sprawdzenia w formie używanej w pliku parametrów. Klucz może dotyczyć pliku lub bazy danych. |

Tabela 30. Parametry `getFieldTypes` (kontynuacja)

| Parametr                 | Opis   |
|--------------------------|--|
| <code>field_count</code> | Liczba zmiennych do sprawdzenia. Ta długość musi być mniejsza lub równa długości tablicy <code>field_types</code> .  |
| <code>field_types</code> | Tablica o długości przynajmniej <code>field_count</code> , która otrzymuje typy zmiennych. Typy pierwszych zmiennych <code>field_count</code> w danych wejściowych lub wyjściowych są kopiowane do tablicy. Jeśli wartość <code>field_count</code> jest większa niż rzeczywista liczba zmiennych, dodatkowe elementy w tablicy pozostają niezdefiniowane. Wartości dla typów zmiennych są wymienione w poniższej tabeli Typy danych. |

## Typy danych

`field_types` musi być jednym z typów danych przedstawionych w poniższej tabeli.

Tabela 31. Typy danych

| Typ       | Interpretacja   | Standardowy deklaratorem C |
|-----------|---|----------------------------|
| STRING    | Łańcuch znaków UTF-8 przerywany wartością null.                             | <code>const char*</code>   |
| INTEGER   | 32-bitowa liczba całkowita ze znakiem.                                      | <code>int</code>           |
| LONG      | 64-bitowa liczba całkowita ze znakiem.                                      | <code>long long</code>     |
| REAL      | 64-bitowa zmiennopozycyjna.   | <code>double</code>        |
| TIME      | 64-bitowa liczba całkowita ze znakiem (sekundy od północy).                 | <code>long long</code>     |
| DATE      | 64-bitowa liczba całkowita ze znakiem (sekundy od północy 1 stycznia 1970). | <code>long long</code>     |
| TIMESTAMP | 64-bitowa liczba całkowita ze znakiem (sekundy od północy 1 stycznia 1970). | <code>long long</code>     |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 32. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

## setAlternativeInput

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

Zastępuje źródło wejściowe pliku alternatywnym źródłem wejściowym.

Tabela 33. Parametry `setAlternativeInput`

| Parametr    | Opis   |
|-------------|--|
| handle      | Uchwyt obrazu.   |
| key         | Nazwa oryginalnego źródła wejściowego w formie używanej w pliku parametrów. Źródło wejściowe musi być plikiem, aby klucz zawsze miał formę <i>fileN</i> dla liczb całkowitych $N \geq 0$ .   |
| field_count | Liczba zmiennych w danych wejściowych. Wartość musi odpowiadać dokładnie liczbie zmiennych w oryginalnym pliku wejściowym lub wywołanie zakończy się niepowodzeniem.   |
| field_types | Tablica typów o długości przynajmniej <code>field_count</code> . Wartość <code>field_types[i]</code> to typ w <i>i</i> -tej zmiennej i musi być jedną z wartości wymienionych w poniższej tabeli Typy danych. Typy muszą być kompatybilne z typami w oryginalnym pliku wejściowym lub wywołanie zakończy się niepowodzeniem.   |
| iterator    | <p>Funkcja, która generuje alternatywne dane wejściowe. Funkcja jest stosowana dla swojego argumentu w następujący sposób:</p> <pre>void** row = iterator(arg);</pre> <p>Funkcja jest wywoływana podczas wykonywania (wewnątrz wywołania do <code>clemrtl_execute</code>) i jest wywoływana raz dla każdego rekordu wejściowego. Wartość zwrotna NULL wskazuje koniec danych wejściowych. W takim przypadku funkcja nie jest wywoływana ponownie i wykonywanie zostanie zatrzymane. W przeciwnym wypadku wynikiem jest tablica danych o długości przynajmniej <code>field_count</code>, gdzie <code>row[i]</code> podaje wartość w <i>i</i>-tej zmiennej. Wartością może być NULL, w przeciwnym razie musi to być wskaźnik do układu odniesienia, którego typ jest określony powiązaną wartością <code>field_types[i]</code>. Wskaźniki muszą pozostać prawidłowe do momentu następnego wywołania iteratora lub do końca wykonywania, jeśli wykonywanie zostanie zakończone wcześniej.</p> |
| arg         | Argument nieprzezroczysty, który jest przekazywany do iteratora podczas każdego wywołania.   |

## Typy danych

Lista `field_types` musi być jedną z najbardziej ograniczonych list typów. Więcej informacji można znaleźć w temacie “`getFieldTypes`” na stronie 28.

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 34. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli typy zmiennych nie są kompatybilne z typami w oryginalnym pliku wejściowym lub liczba zmiennych nie odpowiada dokładnie liczbie zmiennych w oryginalnym pliku wejściowym.

## 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);

```

Zastępuje wyjściowy element docelowy pliku alternatywnym wyjściowym elementem docelowym.

Tabela 35. Parametry `setAlternativeOutput`

| Parametr    | Opis  |
|-------------|---|
| handle      | Uchwyt obrazu.  |
| key         | Nazwa oryginalnego wyjściowego elementu docelowego w formie używanej w pliku parametrów. Wyjściowy element docelowy musi być plikiem, aby klucz zawsze miał formę <i>fileN</i> dla liczb całkowitych $N \geq 0$ .   |
| field_count | Liczba zmiennych w danych wyjściowych. Wartość musi odpowiadać dokładnie liczbie zmiennych w oryginalnym pliku wyjściowym lub wywołanie zakończy się niepowodzeniem.  |
| field_types | Tablica typów o długości przynajmniej <code>field_count</code> . Wartość <code>field_types[i]</code> to typ w <i>i</i> -tej zmiennej i musi być jedną z wartości wymienionych w poniższej tabeli Typy danych. Typy muszą być kompatybilne z typami w oryginalnym pliku wyjściowym lub wywołanie zakończy się niepowodzeniem.  |
| iterator    | Funkcja, która używa danych wyjściowych obrazu. Funkcja jest stosowana dla swojego argumentu i wiersza danych w następujący sposób:<br><code>iterator(arg, row);</code><br><br>Funkcja jest wywoływana podczas wykonywania (wewnątrz wywołania do <code>clemrtl_execute</code> ) i jest wywoływana raz dla każdego wiersza wyników wygenerowanego przez obraz. Wartość <code>row</code> równa <code>NULL</code> wskazuje na koniec danych wyjściowych, po czym funkcja nie jest już wywoływana. Aplikacja nie powinna polegać na tym końcowym wywołaniu i powinna opróżnić i zamknąć wszystkie zasoby zewnętrzne itp., gdy wykonywanie zostało przerwane. W przeciwnym wypadku <code>row</code> to tablica danych o długości przynajmniej <code>field_count</code> , gdzie <code>row[i]</code> podaje wartość w <i>i</i> -tej zmiennej. Wartością może być <code>NULL</code> , w przeciwnym razie musi to być wskaźnik do układu odniesienia, którego typ jest określony powiązaną wartością <code>field_types[i]</code> . Funkcja musi kopiować wszystkie wartości danych, których potrzebuje, ponieważ pamięć może nie zostać zachowana po zwróceniu wywołania. |
| arg         | Argument nieprzezroczysty, który jest przekazywany do iteratora podczas każdego wywołania.  |

## Typy danych

Lista `field_types` musi być jedną z najbardziej ograniczonych list typów. Więcej informacji można znaleźć w temacie “`getFieldTypes`” na stronie 28.

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 36. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_FAIL  | Niepowodzenie bez dostępnych bardziej szczegółowych informacji. |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli typy zmiennych nie są kompatybilne z typami w oryginalnym pliku wyjściowym lub liczba zmiennych nie odpowiada dokładnie liczbie zmiennych w oryginalnym pliku wyjściowym.

## execute

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

Wykonuje obraz.

Tabela 37. Parametry execute

| Parametr | Opis           |
|----------|----------------|
| handle   | Uchwyt obrazu. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 38. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Jeśli obraz nie został przygotowany, jest najpierw przygotowany przy pomocy bieżących wartości parametrów. Operacja kończy się niepowodzeniem, jeśli uchwyt obrazu jest nieprawidłowy lub wystąpi błąd podczas przygotowania lub wykonywania. Wywołanie nie jest zwracane do momentu zakończenia wykonywania.

## prepare

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

Przygotowuje obraz do wykonania.

Tabela 39. Parametry prepare

| Parametr | Opis           |
|----------|----------------|
| handle   | Uchwyt obrazu. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 40. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Operacja kończy się niepowodzeniem, jeśli uchwyt obrazu jest nieprawidłowy lub wystąpi błąd podczas przygotowania.

Obraz musi być przygotowany przed wykonaniem. Przygotowywanie obrazu utrwała wartości parametrów w obrazie. Funkcja `clemrtl_execute()` automatycznie przygotowuje obraz przy każdym wywołaniu, jeśli nie został już przygotowany. Użycie funkcji `clemrtl_prepare()` do przygotowania obrazu jest przydatne, jeśli obraz ma być wykonywany wiele razy z tymi samymi wartościami parametrów, i może to znacznie poprawić wydajność. Po przygotowaniu obrazu wszelkie zmiany wartości parametrów są ignorowane. Użyj ponownie funkcji `clemrtl_prepare()`, aby aktualizować obraz za pomocą nowych wartości parametrów.

## interrupt

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

Przerywa wykonywanie postępu obrazu.

Tabela 41. Parametry interrupt

| Parametr | Opis           |
|----------|----------------|
| handle   | Uchwyt obrazu. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 42. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Ta operacja nie odnosi skutku, jeśli uchwyt obrazu jest nieprawidłowy lub obraz nie jest wykonywany.

Tę funkcję można bezpiecznie wywołać razem z innym wywołaniem API tego samego uchwytu obrazu.

## getErrorDetail

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

Pobiera szczegółowe informacje o ostatnim błędzie, który wystąpił dla obrazu.

Tabela 43. Parametry getErrorDetail

| Parametr  | Opis  |
|-----------|---|
| handle    | Uchwyt obrazu.  |
| severity  | Otrzymuje kod istotności jako pojedynczy znak:<br><br>I — informacje<br>W — ostrzeżenie<br>E — błąd<br>X — błąd systemowy |
| code      | Otrzymuje numer błędu.  |
| text      | Bufor otrzymujący tekst komunikatu.   |
| text_size | Rozmiar bufora tekstu.  |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 44. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Wyniki będą niewiarygodne, jeśli uchwyt obrazu jest nieprawidłowy. Tekst komunikatu jest przycinany, jeśli niezbędne jest dopasowanie do bufora tekstu.

Jeśli wywołanie API kończy się niepowodzeniem z błędem CLEMRTL\_ERROR i błąd nie jest powiązany z konkretnym uchwycem obrazu, przekazanie wartości 0 jako uchwytu obrazu spowoduje pobranie szczegółów ostatniego błędu niepowiązanego z konkretnym obrazem, ale jest to zawodne w środowisku wielowątkowym.

## setReportHandler

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

Instaluje procedurę wywołania zwrotnego raportu dla obrazu.

Tabela 45. Parametry setReportHandler

| Parametr | Opis   |
|----------|--|
| handle   | Uchwyt obrazu.   |
| proc     | Procedura wywołania zwrotnego raportu.                   |
| data     | Dane określone przez użytkownika do wywołania zwrotnego. |

Zwraca jeden z kodów statusu przedstawionych w poniższej tabeli.

Tabela 46. Kody statusów

| Wynik         | Opis  |
|---------------|---|
| CLEMRTL_OK    | Powodzenie.   |
| CLEMRTL_ERROR | Niepowodzenie z dostępnymi bardziej szczegółowymi informacjami. |

Procedura wywołania zwrotnego ma następujący typ:

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

Tabela 47. Parametry procedury wywołania zwrotnego

| Parametr | Opis  |
|----------|---|
| data     | Dane określone przez użytkownika przekazywane do clemrtl_setReportHandler().                                    |
| severity | Kod istotności jako pojedynczy znak:<br><br>I — informacje<br>W — ostrzeżenie<br>E — błąd<br>X — błąd systemowy |
| code     | Numer komunikatu.   |
| text     | Tekst komunikatu.   |



Procedura jest stosowana do każdego komunikatu, gdy jest odbierany. Ustawienie procedury na wartość NULL usuwa każdy istniejący uchwyt zainstalowany na obrazie.



---

## Uwagi

Niniejsza publikacja została przygotowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych. Materiał ten jest również dostępny w IBM w innych językach. Jednakże w celu uzyskania dostępu do takiego materiału istnieje konieczność posiadania egzemplarza produktu w takim języku.

Produktów, usług lub opcji opisywanych w tym dokumencie IBM nie musi oferować we wszystkich krajach. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem że nie narusza to praw własności intelektualnej IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Pisemne zapytania w sprawie licencji można przysyłać na adres:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
U.S.A.*

Zapytania dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub wysłać je na piśmie na adres:

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

INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŹNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy drukarskie. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkownika i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjodawcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
U.S.A.*

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Dane dotyczące wydajności i cytowane przykłady zostały przedstawione jedynie w celu zobrazowania sytuacji. Faktyczne wyniki dotyczące wydajności mogą się różnić w zależności do konkretnych warunków konfiguracyjnych i operacyjnych.

Informacje dotyczące produktów innych podmiotów niż IBM zostały uzyskane od dostawców tych produktów, z ich publicznych ogłoszeń lub innych dostępnych publicznie źródeł. IBM nie testował tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennej pracy. W celu kompleksowego ich zilustrowania, podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk jest całkowicie przypadkowe.

---

## Znaki towarowe

IBM, logo IBM i [ibm.com](http://ibm.com) są znakami towarowymi lub zastrzeżonymi znakami towarowymi International Business Machines Corp. zarejestrowanymi w wielu systemach prawnych na całym świecie. Pozostałe nazwy produktów i usług mogą być znakami towarowymi IBM lub innych przedsiębiorstw. Aktualna lista znaków towarowych IBM dostępna jest w serwisie WWW IBM, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, logo Adobe, PostScript oraz logo PostScript są znakami towarowymi lub zastrzeżonymi znakami towarowymi Adobe Systems Incorporated w Stanach Zjednoczonych i/lub w innych krajach.

Intel, logo Intel, Intel Inside, logo Intel Inside, Intel Centrino, logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium i Pentium są znakami towarowymi lub zastrzeżonymi znakami towarowymi Intel Corporation lub przedsiębiorstw podporządkowanych Intel Corporation w Stanach Zjednoczonych i w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Microsoft, Windows, Windows NT oraz logo Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych.

---

## Warunki dotyczące dokumentacji produktu

Zezwolenie na korzystanie z tych publikacji jest przyznawane na poniższych warunkach.

### Zakres stosowania

Niniejsze warunki stanowią uzupełnienie warunków używania serwisu WWW IBM.

### Użytek osobisty

Użytkownik ma prawo kopiować te publikacje do własnego, niekomercyjnego użytku pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa dystrybuować ani wyświetlać tych publikacji czy ich części, ani też wykonywać na ich podstawie prac pochodnych bez wyraźnej zgody IBM.

### Użytek służbowy

Użytkownik ma prawo kopiować te publikacje, dystrybuować je i wyświetlać wyłącznie w ramach przedsiębiorstwa Użytkownika pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa wykonywać na podstawie tych publikacji ani ich fragmentów prac pochodnych, kopiować ich, dystrybuować ani wyświetlać poza przedsiębiorstwem Użytkownika bez wyraźnej zgody IBM.

### Prawa

Z wyjątkiem zezwoleń wyraźnie udzielonych w niniejszym dokumencie, nie udziela się jakichkolwiek innych zezwoleń, licencji ani praw, wyraźnych czy domniemanych, odnoszących się do tych publikacji czy jakichkolwiek informacji, danych, oprogramowania lub innej własności intelektualnej, o których mowa w niniejszym dokumencie.

IBM zastrzega sobie prawo do anulowania zezwolenia przyznanego w niniejszym dokumencie w każdej sytuacji, gdy, według uznania IBM, korzystanie z tych publikacji jest szkodliwe dla IBM lub jeśli IBM uzna, że warunki niniejszego dokumentu nie są przestrzegane.

Użytkownik ma prawo pobierać, eksportować lub reeksportować niniejsze informacje pod warunkiem zachowania bezwzględnej i pełnej zgodności z obowiązującym prawem i przepisami, w tym ze wszelkimi prawami i przepisami eksportowymi Stanów Zjednoczonych.

IBM NIE UDZIELA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, DOTYCZĄCYCH TREŚCI TYCH PUBLIKACJI. PUBLIKACJE TE SĄ DOSTARCZANE W STANIE, W JAKIM SIĘ ZNAJDUJĄ ("AS-IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁACZA SIĘ), WYRAŹNYCH CZY DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU CZY NIENARUSZANIA PRAW OSÓB TRZECICH.



# Indeks

## A

Analiza predykcijna IBM SPSS umożliwiająca osadzenie  
API 19

API  
alternatywne dane wejściowe 29  
alternatywne dane wyjściowe 30  
informacje 21  
kody błędów 21  
kody statusów 21  
liczba zmiennych 28  
opcje wykonywania 23, 24  
przegląd 13, 21  
setReportHandler 34  
typy pól 28

## B

Biblioteka oprogramowania środowiska wykonawczego (CLEMRTL) 13

## C

closeImage  
Informacje o funkcjach API 25  
Przegląd procesu API 21

## D

deinstalacja produktu  
IBM SPSS Modeler Solution Publisher  
Runtime 18  
dokumentacja 3

## E

enumerateParameters  
Informacje o funkcjach API 26  
Przegląd procesu API 21

EPM  
API 19

execute  
Informacje o funkcjach API 32  
Przegląd procesu API 21

## F

Funkcje API  
closeImage 21, 25  
enumerateParameters 21, 26  
execute 21, 32  
getErrorDetail 21, 33  
getFieldCount 21, 28  
getFieldTypes 21, 28  
getOption 21, 23  
getParameter 21, 27  
inicjowanie 21, 22  
interrupt 21, 33

Funkcje API (*kontynuacja*)  
openImage 21, 25  
prepare 21, 32  
setAlternativeInput 21, 29  
setAlternativeOutput 21, 30  
setLogFile 21, 24  
setOption 21, 24  
setParameter 21, 27  
setReportHandler 21

## G

getErrorDetail  
Informacje o funkcjach API 33  
Przegląd procesu API 21

getFieldCount  
Informacje o funkcjach API 28  
Przegląd procesu API 21

getFieldTypes  
Informacje o funkcjach API 28  
Przegląd procesu API 21

getOption  
Informacje o funkcjach API 23  
Przegląd procesu API 21

getParameter  
Informacje o funkcjach API 27  
Przegląd procesu API 21

## I

IBM SPSS Modeler 1  
dokumentacja 3

IBM SPSS Modeler Server 1

IBM SPSS Modeler Solution Publisher 7, 8, 19  
osadzanie w aplikacjach 13  
przegląd 7

IBM SPSS Modeler Solution Publisher  
Runtime  
deinstalacja produktu 18  
instalacja 15  
katalog tymczasowy 17  
uruchamianie 17  
wymagania systemowe 15

inicjowanie  
flags 22  
Informacje o funkcjach API 22  
Przegląd procesu API 21

initialise\_ext  
flags 22  
Informacje o funkcjach API 22

interrupt  
Informacje o funkcjach API 33  
Przegląd procesu API 21

## K

katalog plików tymczasowych  
IBM SPSS Modeler Solution Publisher  
Runtime 17

kody błędów 21  
kody statusów 21

## O

openImage  
Informacje o funkcjach API 25  
Przegląd procesu API 21

otwieranie  
strumienie 25

## P

parametry  
Parametry obrazu API 26, 27  
parametry wykonywania strumienia 12  
Pliki parametrów API 25  
Procedury wywołania zwrotnego API 26

Plik .par 8  
Plik .pim 8  
pliki dzienników 24

prepare  
Informacje o funkcjach API 32  
Przegląd procesu API 21

przykłady  
Podręcznik zastosowań 3  
przegląd 4  
przykłady aplikacji 3  
publikowanie strumieni  
IBM SPSS Modeler Solution Publisher 7, 8

## R

raportowanie błędów  
Informacje o funkcjach API 24, 33, 34

## S

setAlternativeInput  
Informacje o funkcjach API 29  
Przegląd procesu API 21

setAlternativeOutput  
Informacje o funkcjach API 30  
Przegląd procesu API 21

setLogFile  
Informacje o funkcjach API 24  
Przegląd procesu API 21

setOption  
Informacje o funkcjach API 24  
Przegląd procesu API 21

setParameter  
Informacje o funkcjach API 27  
Przegląd procesu API 21

setReportHandler  
Informacje o funkcjach API 34  
Przegląd procesu API 21

strefa czasowa  
ustawianie za pomocą API 22

## Ś

- Środowisko wykonawcze
  - deinstalacja produktu 18
  - Instalowanie w systemie Windows 15
  - katalog tymczasowy 17
  - opcje 11
  - UNIX - instalowanie 15
  - uruchamianie 17

## U

- UNIX
  - instalowanie produktu IBM SPSS Modeler Solution Publisher Runtime 15
- ustawienia lokalne
  - ustawianie za pomocą API 22

## W

- wdrażanie 7, 19
- Węzeł IBM SPSS Modeler Solution Publisher 19
- Węzeł Publikuj 19
- węzły wyników 7, 19
- wykonywanie strumieni
  - używanie produktu IBM SPSS Modeler Solution Publisher 11, 12

## Z

- zamykanie
  - strumienie 25







Drukowane w USA