

*IBM SPSS Modeler 17.1 Solution
Publisher*

IBM

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 39.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.ibm.com/ca/fr> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette édition s'applique à la version 17.1.0 d'IBM(r) SPSS(r) Modeler et à toutes les éditions et modifications suivantes, sauf indication contraire dans les nouvelles éditions.

Table des matières

Avis aux lecteurs canadiens	v	Exemple d'application IBM SPSS Modeler Solution Publisher	14
Préface	vii		
Chapitre 1. A propos d'IBM SPSS Modeler	1	Chapitre 4. Installation d'IBM SPSS Modeler Solution Publisher Runtime	15
Produits IBM SPSS Modeler	1	Installation de Runtime	15
IBM SPSS Modeler	1	Configuration requise	15
IBM SPSS Modeler Server	1	Installation sous Windows	16
IBM SPSS Modeler Administration Console	2	Installation sous UNIX	16
IBM SPSS Modeler Batch	2	Dépannage d'une installation	17
IBM SPSS Modeler Solution Publisher	2	Démarrage du programme Runtime	18
Adaptateurs IBM SPSS Modeler Server pour IBM SPSS Collaboration and Deployment Services	2	Changement de répertoire temporaire	18
Editions de IBM SPSS Modeler	2	Désinstallation de IBM SPSS Modeler Solution Publisher Runtime	19
Documentation de IBM SPSS Modeler	3	Procédure de désinstallation sous Windows	19
Documentation de SPSS Modeler Professional	3	Procédure de désinstallation sous UNIX	19
Documentation de SPSS Modeler Premium	4		
Exemples d'application	5	Chapitre 5. API IBM SPSS Embedded Predictive Modeling	21
Dossier Demos	5		
		Chapitre 6. Référence de l'API de la bibliothèque d'exécution IBM SPSS Modeler Solution Publisher	23
Chapitre 2. IBM SPSS Modeler Solution Publisher	7	Introduction	23
Fonctionnement de IBM SPSS Modeler Solution Publisher	7	Présentation du processus d'API	23
Publication de flux	8	Fonctions API	24
		Remarques	39
Chapitre 3. Exécution de flux publiés	11	Marques	40
Utilisation du fichier de paramètres pour personnaliser l'exécution de flux	12	Index	43
Intégration de IBM SPSS Modeler Solution Publisher dans les applications	13		

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Préface

IBM® SPSS Modeler est le puissant utilitaire d'exploration de données de IBM Corp.. SPSS Modeler aide les entreprises et les organismes à améliorer leurs relations avec les clients et les citoyens grâce à une compréhension approfondie des données. A l'aide des connaissances plus précises obtenues par le biais de SPSS Modeler, les entreprises et les organismes peuvent conserver les clients rentables, identifier les opportunités de vente croisée, attirer de nouveaux clients, détecter les éventuelles fraudes, réduire les risques et améliorer les services gouvernementaux.

L'interface visuelle de SPSS Modeler met à contribution les compétences professionnelles de l'utilisateur, ce qui permet d'obtenir des modèles prédictifs plus efficaces et de trouver des solutions plus rapidement. SPSS Modeler offre de nombreuses techniques de modélisation, telles que les algorithmes de prévision, de classification, de segmentation et de détection d'association. Une fois les modèles créés, l'utilisateur peut utiliser IBM SPSS Modeler Solution Publisher pour les remettre aux responsables, où qu'ils se trouvent dans l'entreprise, ou pour les transférer vers une base de données.

A propos d'IBM Business Analytics

Le logiciel IBM Business Analytics propose des informations complètes, cohérentes et précises auxquelles les preneurs de décisions peuvent se fier pour améliorer les performances de leur entreprise. Un porte-feuilles étendu de veille économique, d'analyses prédictives, de gestion des performances et de stratégie financières et d'applications analytiques vous offre des informations claires, immédiates et décisionnelles sur les performances actuelles et vous permet de prévoir les résultats futurs. Ce logiciel intègre des solutions dédiées à l'industrie, des pratiques éprouvées et des services professionnels qui permettent aux organisations de toute taille de maximiser leur productivité, d'automatiser leurs décisions sans risque et de proposer de meilleurs résultats.

Ce porte-feuilles intègre le logiciel IBM SPSS Predictive Analytics qui aide les organisations à prévoir les événements à venir et à réagir en fonction des informations afin d'améliorer leurs résultats. Les clients de l'industrie du commerce, de l'éducation et des administrations du monde entier font confiance à la technologie IBM SPSS qui offre un avantage concurrentiel en attirant et fidélisant les clients et en améliorant la base de données de la clientèle tout en diminuant la fraude et en réduisant les risques. En utilisant le logiciel IBM SPSS dans leurs opérations quotidiennes, les organisations deviennent des entreprises prédictives, capables de diriger et d'automatiser les décisions pour répondre aux objectifs commerciaux et obtenir un avantage concurrentiel mesurable. Pour des informations supplémentaires ou pour joindre un représentant, consultez <http://www.ibm.com/spss>.

Assistance technique

L'assistance technique est disponible pour les clients du service de maintenance. Les clients peuvent contacter l'assistance technique pour obtenir de l'aide concernant l'utilisation des produits IBM Corp. ou l'installation dans l'un des environnements matériels pris en charge. Pour contacter l'assistance technique, rendez-vous sur le site Web IBM Corp. à l'adresse <http://www.ibm.com/support>. Lorsque vous contactez l'assistance technique, soyez prêt à indiquer votre identité, le nom de votre société et votre contrat d'assistance.

Chapitre 1. A propos d'IBM SPSS Modeler

IBM SPSS Modeler est un ensemble d'outils d'exploration de données qui vous permet de développer rapidement, grâce à vos compétences professionnelles, des modèles prédictifs et de les déployer dans des applications professionnelles afin de faciliter la prise de décision. Conçu autour d'un modèle confirmé, le modèle CRISP-DM, IBM SPSS Modeler prend en charge l'intégralité du processus d'exploration de données, des données à l'obtention de meilleurs résultats commerciaux.

IBM SPSS Modeler propose différentes méthodes de modélisation issues des domaines de l'apprentissage automatique, de l'intelligence artificielle et des statistiques. Les méthodes disponibles dans la palette Modélisation vous permettent d'extraire de nouvelles informations de vos données et de développer des modèles prédictifs. Chaque méthode possède ses propres avantages et est donc plus adaptée à certains types de problème spécifiques.

Il est possible d'acquérir SPSS Modeler comme produit autonome ou de l'utiliser en tant que client en combinaison avec SPSS Modeler Server. Plusieurs autres options sont également disponibles, telles que décrites dans les sections suivantes. Pour plus d'informations, voir <http://www.ibm.com/software/analytics/spss/products/modeler/>.

Produits IBM SPSS Modeler

La famille des produits IBM SPSS Modeler et les logiciels associés sont composés des éléments suivants.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- Adaptateurs IBM SPSS Modeler Server pour IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler est une version complète du produit que vous installez et exécutez sur votre ordinateur personnel. Pour obtenir de meilleures performances lors du traitement de jeux de données volumineux, vous pouvez exécuter SPSS Modeler en mode local, comme produit autonome, ou l'utiliser en mode réparti, en association avec IBM SPSS Modeler Server.

Avec SPSS Modeler, vous pouvez créer des modèles prédictifs précis rapidement et de manière intuitive, sans aucune programmation. L'interface visuelle unique vous permet de visualiser facilement le processus d'exploration de données. Grâce aux analyses avancées intégrées au produit, vous pouvez découvrir des motifs et tendances masqués dans vos données. Vous pouvez modéliser les résultats et comprendre les facteurs qui les influencent, afin d'exploiter les opportunités commerciales et de réduire les risques.

SPSS Modeler est proposé dans deux éditions : SPSS Modeler Professional et SPSS Modeler Premium. Pour plus d'informations, voir la rubrique «Editions de IBM SPSS Modeler», à la page 2.

IBM SPSS Modeler Server

Grâce à une architecture client/serveur, SPSS Modeler adresse les demandes d'opérations très consommatrices de ressources à un logiciel serveur puissant. Il offre ainsi des performances accrues sur des jeux de données plus volumineux.

SPSS Modeler Server est un produit avec licence distincte qui s'exécute en permanence en mode d'analyse réparti sur un hôte de serveur en combinaison avec une ou plusieurs installations de IBM SPSS Modeler. Ainsi, SPSS Modeler Server fournit des performances supérieures sur de grands jeux de données car les opérations nécessitant beaucoup de mémoire peuvent être effectuées sur le serveur sans télécharger de données sur l'ordinateur client. IBM SPSS Modeler Server prend également en charge l'optimisation SQL et propose des capacités de modélisation dans la base de données pour des performances et une automatisation améliorées.

IBM SPSS Modeler Administration Console

Modeler Administration Console est une application graphique permettant de gérer de nombreuses options de SPSS Modeler Server qui peuvent également être configurées au moyen d'un fichier d'options. Cette application offre une interface utilisateur sous forme de console permettant de surveiller et de configurer les installations SPSS Modeler Server ; elle est disponible gratuitement pour les clients actuels de SPSS Modeler Server. L'application ne peut être installée que sur des ordinateurs Windows ; en revanche, elle peut administrer un serveur installé sur n'importe quelle plate-forme prise en charge.

IBM SPSS Modeler Batch

Alors que l'exploration de données est généralement un processus interactif, il est également possible d'exécuter SPSS Modeler à partir d'une ligne de commande sans recourir à l'interface utilisateur graphique. Par exemple, vous pouvez avoir des tâches longue durée ou répétitives à exécuter sans intervention de l'utilisateur. SPSS Modeler Batch est une version spécifique du produit qui prend en charge toutes les capacités d'analyse de SPSS Modeler sans avoir besoin d'accéder à l'interface utilisateur standard. SPSS Modeler Server est requis pour utiliser SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher est un outil qui permet de créer une version « packagée » d'un flux SPSS Modeler qui peut être exécutée par un moteur Runtime externe ou intégrée dans une application externe. Ainsi, vous pouvez publier et déployer des flux SPSS Modeler complets dans des environnements où SPSS Modeler n'est pas installé. SPSS Modeler Solution Publisher est fourni avec le service IBM SPSS Collaboration and Deployment Services - Scoring et nécessite une licence distincte. Avec cette licence, vous recevez SPSS Modeler Solution Publisher Runtime qui vous permet d'exécuter les flux publiés.

Pour plus d'informations sur SPSS Modeler Solution Publisher, reportez-vous à la documentation d'IBM SPSS Collaboration and Deployment Services. Le Knowledge Center d'IBM SPSS Collaboration and Deployment Services contient des sections intitulées "IBM SPSS Modeler Solution Publisher" et "IBM SPSS Analytics Toolkit."

Adaptateurs IBM SPSS Modeler Server pour IBM SPSS Collaboration and Deployment Services

Différents adaptateurs pour IBM SPSS Collaboration and Deployment Services sont disponibles et permettent à SPSS Modeler et SPSS Modeler Server d'interagir avec un référentiel IBM SPSS Collaboration and Deployment Services. Ainsi, un flux SPSS Modeler déployé sur le référentiel peut être partagé par différents utilisateurs ou peut être accessible depuis l'application client léger IBM SPSS Modeler Advantage. Installez l'adaptateur sur le système qui héberge le référentiel.

Editions de IBM SPSS Modeler

SPSS Modeler est disponible dans les éditions suivantes.

SPSS Modeler Professional

SPSS Modeler Professional offre tous les outils nécessaires à l'utilisation de la plupart des types de données structurées, tels que les comportements et interactions suivis dans les systèmes CRM, les

caractéristiques sociodémographiques, les comportements d'achat et les données de vente.

SPSS Modeler Premium

SPSS Modeler Premium est un produit avec licence distincte qui étend le champ d'applications de SPSS Modeler Professional afin de pouvoir traiter des données spécialisées telles que celles utilisées pour les analyses d'entités ou les réseaux sociaux ainsi que des données de texte non structurées. SPSS Modeler Premium comprend les composants suivants :

IBM SPSS Modeler Entity Analytics ajoute une dimension supplémentaire aux analyses prédictives IBM SPSS Modeler. Alors que les analyses prédictives essaient de prévoir les comportements futurs à partir de données passées, les analyses d'entités se concentrent sur l'amélioration de la cohérence des données actuelles en résolvant les conflits d'identités dans les enregistrements eux-mêmes. Une identité peut être celle d'un individu, d'une organisation, d'un objet ou d'une autre entité pour laquelle une ambiguïté peut exister. La résolution d'identité peut être vitale dans de nombreux domaines, y compris la gestion de la relation client, la détection de la fraude, le blanchiment d'argent et la sécurité nationale et internationale.

IBM SPSS Modeler Social Network Analysis transforme les informations sur les relations en champs qui caractérisent le comportement social des individus et des groupes. Grâce aux données qui décrivent les relations qui sous-tendent les réseaux sociaux, IBM SPSS Modeler Social Network Analysis identifie les chefs sociaux qui influencent le comportement des autres individus du réseau. De plus, il est possible de déterminer les individus qui sont le plus influencés par les autres participants du réseau. En combinant ces résultats avec d'autres mesures, il est possible de créer des profils détaillés des individus sur lesquels baser vos modèles prédictifs. Les modèles qui contiennent ces informations sociales seront plus efficaces que les modèles qui en sont dépourvus.

IBM SPSS Modeler Text Analytics utilise des technologies linguistiques avancées et le traitement du langage naturel pour traiter rapidement une large variété de données textuelles non structurées, en extraire les concepts clés et les organiser pour les regrouper dans des catégories. Les concepts extraits et les catégories peuvent ensuite être combinés aux données structurées existantes, telles que les données démographiques, et appliqués à la modélisation grâce à la gamme complète d'outils d'exploration de données de IBM SPSS Modeler, afin de favoriser une prise de décision précise et efficace.

Documentation de IBM SPSS Modeler

Une documentation au format d'aide en ligne est disponible dans le menu Aide de SPSS Modeler. Cela comprend la documentation de SPSS Modeler, SPSS Modeler Server, ainsi que le guide des applications (appelé également tutoriel), et d'autres documents connexes.

La documentation complète de chaque produit (y compris les instructions d'installation) au format PDF est disponible dans le dossier *\Documentation* de chaque DVD de produit. Les documents d'installation peuvent également être téléchargés depuis le Web sur le site <http://www.ibm.com/support/docview.wss?uid=swg27043831>.

La documentation sous ces deux formats est également disponible depuis le site SPSS Modeler Knowledge Center à l'adresse http://www-01.ibm.com/support/knowledgecenter/SS3RA7_17.0.0.0.

Documentation de SPSS Modeler Professional

La suite de documentation SPSS Modeler Professional (à l'exception des instructions d'installation) est la suivante.

- **Guide d'utilisation d'IBM SPSS Modeler.** Introduction générale à SPSS Modeler : création de flux de données, traitement des valeurs manquantes, création d'expressions CLEM, utilisation des projets et des rapports et regroupement des flux pour le déploiement dans IBM SPSS Collaboration and Deployment Services, des applications prédictives ou IBM SPSS Modeler Advantage.

- **Noeuds de source, d'exécution et de sortie d'IBM SPSS Modeler.** Descriptions de tous les noeuds utilisés pour lire, traiter et renvoyer les données de sortie dans différents formats. En pratique, cela signifie tous les noeuds autres que les noeuds de modélisation.
- **Noeuds modélisation d'IBM SPSS Modeler.** Descriptions de tous les noeuds utilisés pour créer des modèles d'exploration de données. IBM SPSS Modeler propose différentes méthodes de modélisation issues des domaines de l'apprentissage automatique, de l'intelligence artificielle et des statistiques.
- **Guide des algorithmes d'IBM SPSS Modeler.** Descriptions des fondements mathématiques des méthodes de modélisation utilisées dans IBM SPSS Modeler. Ce guide est disponible au format PDF uniquement.
- **Guide des applications IBM SPSS Modeler.** Les exemples de ce guide fournissent des introductions brèves et ciblées aux méthodes et techniques de modélisation. Une version en ligne de ce guide est également disponible dans le menu Aide. Pour plus d'informations, voir la rubrique «Exemples d'application», à la page 5.
- **Guide de génération de scripts Python et d'automatisation IBM SPSS Modeler.** Ce manuel fournit des informations sur l'automatisation du système via des scripts Python, notamment sur les propriétés pouvant être utilisées pour manipuler les noeuds et les flux.
- **Guide de déploiement d'IBM SPSS Modeler.** Informations sur l'exécution des scénarios et des flux IBM SPSS Modeler comme étapes des travaux d'exécution sous IBM SPSS Collaboration and Deployment Services Deployment Manager.
- **Guide du développeur IBM SPSS Modeler CLEF.** CLEF permet d'intégrer des programmes tiers tels que des programmes de traitement de données ou des algorithmes de modélisation en tant que noeuds dans IBM SPSS Modeler.
- **Guide d'exploration de base de données IBM SPSS Modeler.** Informations sur la manière de tirer parti de la puissance de votre base de données pour améliorer les performances et étendre la gamme des capacités d'analyse via des algorithmes tiers.
- **Guide des performances et d'administration d'IBM SPSS Modeler Server.** Informations sur le mode de configuration et d'administration de IBM SPSS Modeler Server.
- **Guide d'utilisation de la console d'administration d'IBM SPSS Modeler.** Informations concernant l'installation et l'utilisation de l'interface utilisateur de la console permettant de surveiller et de configurer IBM SPSS Modeler Server. La console est implémentée en tant que plug-in à l'application Deployment Manager.
- **Guide CRISP-DM d'IBM SPSS Modeler.** Guide détaillé sur l'utilisation de la méthodologie CRISP-DM pour l'exploration de données avec SPSS Modeler
- **Guide d'utilisation d'IBM SPSS Modeler Batch.** Guide complet sur l'utilisation de IBM SPSS Modeler en mode de traitement par lots, avec des détails sur l'exécution en mode de traitement par lots et les arguments de ligne de commande. Ce guide est disponible au format PDF uniquement.

Documentation de SPSS Modeler Premium

La suite de documentation SPSS Modeler Premium (à l'exception des instructions d'installation) est la suivante.

- **IBM SPSS Modeler Entity Analytics - Guide d'utilisation.** Informations sur l'utilisation des analyses d'entités avec SPSS Modeler, notamment l'installation et la configuration du référentiel, les noeuds d'analyses d'entités et les tâches administratives.
- **IBM SPSS Modeler Social Network Analysis - Guide d'utilisation.** Guide sur l'exécution des analyses de réseaux sociaux avec SPSS Modeler, y compris les analyses de groupe et analyses de diffusion.
- **SPSS Modeler Text Analytics - Guide d'utilisation.** Informations sur l'utilisation des analyses de texte avec SPSS Modeler, notamment sur les noeuds Text Mining, l'espace de travail interactif, les modèles et d'autres ressources.

Exemples d'application

Tandis que les outils d'exploration de données de SPSS Modeler peuvent vous aider à résoudre une grande variété de problèmes métier et organisationnels, les exemples d'application fournissent des introductions brèves et ciblées aux méthodes et aux techniques de modélisation. Les jeux de données utilisés ici sont beaucoup plus petits que les énormes entrepôts de données gérés par certains Data miners, mais les concepts et les méthodes impliqués doivent pouvoir être adaptés à des applications réelles.

Vous pouvez accéder aux exemples en cliquant **Exemples d'application** dans le menu Aide de SPSS Modeler. Les fichiers de données et les flux d'échantillons sont installés dans le dossier *Demos*, sous le répertoire d'installation du produit. Pour plus d'informations, voir la rubrique «Dossier Demos».

Exemples de modélisation de bases de données. Consultez les exemples dans le manuel *Guide d'exploration de base de données IBM SPSS Modeler*.

Exemples de génération de scripts. Consultez les exemples dans le manuel *Guide de génération de scripts et d'automatisation IBM SPSS Modeler*.

Dossier Demos

Les fichiers de données et les flux d'échantillons utilisés avec les exemples d'application sont installés dans le dossier *Demos*, sous le répertoire d'installation du produit. Ce dossier est également accessible à partir du groupe de programmes IBM SPSS Modeler dans le menu Démarrer de Windows ou en cliquant sur *Demos* dans la liste des répertoires récents de la boîte de dialogue Ouverture de fichier.

Chapitre 2. IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher est un puissant outil qui permet d'intégrer les résultats d'exploration de données dans votre processus métier afin de résoudre des problèmes du monde réel. A l'aide d'IBM SPSS Modeler Solution Publisher, vous pouvez créer une version complète d'un flux pouvant être exécutée par un moteur Runtime externe ou intégrée dans une application externe. De cette manière, vous pouvez déployer vos flux de modélisation de données dans un environnement de production afin de prendre en charge vos processus métier quotidiens et aider les décideurs de l'entreprise grâce aux connaissances issues de l'exploration des données.

A l'aide d'IBM SPSS Modeler Solution Publisher, vous pouvez non seulement exporter des modèles (au format PMML), mais également publier et déployer des *flux IBM SPSS Modeler complets*. Par conséquent, vous pouvez effectuer la préparation des données, ainsi que des opérations sur les enregistrements et les champs, notamment l'agrégation de données, la sélection d'enregistrements ou le calcul de nouveaux champs, avant de créer des prévisions basées sur un modèle. Vous pouvez ensuite davantage traiter les résultats de modèle avant d'enregistrer les données, le tout en exécutant simplement le flux publié.

Remarque : La publication est réalisée en utilisant l'onglet Publier du noeud d'export approprié. Pour plus d'informations, voir la rubrique «Publication de flux», à la page 8. Les flux enregistrés dans les versions antérieures à 11.0 du produit ont utilisé un noeud Publier distinct. Le noeud est déjà pris en charge pour la compatibilité descendante, mais il n'est plus disponible sur la palette de noeud.

Octroi de licence IBM SPSS Modeler Solution Publisher

Remarque : IBM SPSS Modeler Solution Publisher est fourni avec le service d'évaluation d'IBM SPSS Collaboration and Deployment Services et nécessite une licence distincte. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Lorsque vous vous procurez une licence, vous obtenez une application IBM SPSS Modeler Solution Publisher Runtime distincte permettant d'exécuter des flux publiés. Pour plus d'informations, voir la rubrique «Installation de Runtime», à la page 15.

Fonctionnement de IBM SPSS Modeler Solution Publisher

Le déploiement d'une solution à l'aide d'IBM SPSS Modeler Solution Publisher se déroule en deux phases : *publication* et *exécution* d'un flux.

Publication. Après avoir travaillé sur le processus d'exploration de données, vous obtiendrez un modèle offrant une bonne solution au problème concerné. A ce stade, vous êtes en mesure d'appliquer le modèle à votre processus métier. Lorsque vous publiez un flux, sa description détaillée est inscrite sur le disque (sous la forme d'un fichier *image* et d'un fichier de *paramètres*). Pour plus d'informations, voir la rubrique «Publication de flux», à la page 8.

Execution. Une fois le flux publié, vous pouvez recréer le processus concerné en exécutant ce flux. Pour ce faire, servez-vous de la version autonome de IBM SPSS Modeler Runtime (*modelerrun.exe*) ou développez une application utilisant la bibliothèque IBM SPSS Modeler Runtime pour exécuter le flux. Pour exécuter des flux à l'extérieur de IBM SPSS Modeler (à l'aide de Runtime ou d'une application personnalisée), vous devez en premier lieu installer IBM SPSS Modeler Solution Publisher Runtime. Pour plus d'informations, voir la rubrique «Installation de Runtime», à la page 15.

Publication de flux

La publication des flux s'effectue directement à partir d'IBM SPSS Modeler à l'aide de n'importe quel noeud d'export standard : Base de données, Fichier à plat, Export Statistiques, Export IBM SPSS Data Collection, Export SAS, Excel et Export XML. Le type de noeud d'export détermine le format des résultats à écrire chaque fois que le flux publié est exécuté à l'aide de IBM SPSS Modeler Solution Publisher Runtime ou d'une application externe. A titre d'exemple, pour enregistrer vos résultats dans une base de données chaque fois que le flux publié est exécuté, utilisez un noeud exportation de base de données.

Pour publier un flux

1. Ouvrez ou créez un flux, puis reliez un noeud d'export à la fin du flux.
2. Dans l'onglet Publier du noeud d'exportation, spécifiez un nom racine pour les fichiers publiés (le nom du fichier dont les diverses extensions *.pim*, *.par* et *.xml* seront modifiées).
3. Cliquez sur le bouton **Publier** pour publier le flux ou sélectionnez **Publier le flux** pour publier automatiquement le flux chaque fois que le noeud est exécuté.

Nom publié. Indiquez le nom racine de l'image et des fichiers de paramètres publiés.

- Le **fichier image** (**.pim*) fournit toutes les informations nécessaires à l'exécution par Runtime du flux publié, exactement tel qu'il était au moment de l'exportation. Si vous ne pensez pas devoir changer de paramètres du flux (comme la source des données d'entrée ou le fichier des données de sortie), vous pouvez vous contenter de déployer le fichier image.
- Le **fichier de paramètres** (**.par*) contient des informations configurables sur les sources de données, les fichiers de sortie et les options d'exécution. Pour pouvoir contrôler l'entrée ou la sortie du flux sans republier ce dernier, vous avez besoin du fichier de paramètres et du fichier image.
- Le **fichier de métadonnées** (**.xml*) décrit les entrées et les sorties de l'image et de leurs modèles de données. Il est conçu pour être utilisé par des applications qui incorporent la bibliothèque d'exécution et qui ont besoin de connaître la structure des données d'entrée et de sortie.

Remarque : Ce fichier est produit uniquement si vous sélectionnez la boîte de dialogue **Publier les métadonnées**.

Publication des paramètres. Si nécessaire, vous pouvez inclure les paramètres de flux dans le fichier **.par*. Vous pouvez modifier les valeurs des paramètres de flux lorsque vous exécutez l'image soit en modifiant le fichier **.par* ou en utilisant l'API d'exécution.

Cette case active le bouton **Paramètres**. La boîte de dialogue Publier les paramètres s'affiche lorsque vous cliquez sur ce bouton.

Choisissez les paramètres à inclure dans l'image publiée en sélectionnant la case appropriée dans la colonne **Publier**.

Exécution dans le flux. Indiquez si le flux est publié automatiquement lorsque le noeud est exécuté.

- **Exporter les données.** Exécute le noeud d'export de manière standard, sans publier le flux. (Pour résumer, le noeud s'exécute dans IBM SPSS Modeler comme il le ferait si IBM SPSS Modeler Solution Publisher n'était pas disponible.) Si vous sélectionnez cette option, le flux ne sera publié que lorsque vous le demanderez explicitement en cliquant sur le bouton **Publier** dans la boîte de dialogue du noeud d'export. Vous pouvez également publier le flux en cours à l'aide de l'outil Publier de la barre d'outils ou à l'aide d'un script.
- **Publier le flux.** Publie le flux en vue de le déployer à l'aide de IBM SPSS Modeler Solution Publisher. Sélectionnez cette option pour publier automatiquement le flux chaque fois qu'il est exécuté.

Commentaires

- Si vous comptez exécuter le flux publié avec de nouvelles données ou des données mises à jour, n'oubliez pas que l'ordre des champs du fichier d'entrée doit être identique à celui des champs du fichier d'entrée du noeud source spécifié dans le flux publié.
- Lorsque vous publiez des données vers des applications externes, vous pouvez être amené à filtrer les champs superflus ou à renommer des champs pour respecter les exigences de saisie. Pour ce faire, utilisez le noeud Filtrer avant le noeud d'export.

Chapitre 3. Exécution de flux publiés

La génération d'un jeu de données est l'effet principal de l'exécution d'un flux publié. Les données peuvent être stockées sur un fichier du disque ou inscrites dans une base de données. Les flux publiés ne peuvent pas générer de graphiques, de tableaux, de modèles générés ou de sorties autres que des données.

Les flux publiés à l'aide de IBM SPSS Modeler Solution Publisher peuvent être exécutés en utilisant le programme IBM SPSS Modeler Solution Publisher Runtime. Le programme Runtime, *modelerrun.exe*, est exécuté à partir de la ligne de commande, avec des options indiquant le fichier image et le fichier de paramètres (facultatif) à exécuter. La commande Runtime est la suivante :

```
modelerrun -nobanner -p <fichier de paramètres> -o <options> <fichier image>
```

Les entrées de la commande sont répertoriées dans le tableau ci-après.

Tableau 1. Entrées de commande Runtime

Entrée de commande	Description
-nobanner	supprime le message du texte de démarrage du programme Runtime (facultatif).
<parameter-file>	est le nom du fichier de paramètres (facultatif).
<options>	est une liste séparée par des virgules de paires option=value précisant les options d'exécution du programme Runtime. Les options en question sont décrites plus bas (facultatif). Remarque : Il ne doit pas y avoir d'espaces vides après une virgule.
<image-file>	est le nom du fichier image à exécuter (obligatoire).

Les options d'exécution disponibles sont répertoriées dans le tableau ci-après.

Tableau 2. Options Runtime valides.

Nom	Valeur par défaut	Description
max_file_size	-1	Taille maximale des fichiers enregistrés. La valeur -1 indique qu'il n'y a pas de limite.
max_sql_string_length	2,048	Longueur maximale d'une chaîne importée depuis la base de données avec SQL. Les valeurs de chaîne supérieures à cette longueur sont tronquées sur la droite sans avertissement. La plage valide est comprise entre 1 et 65 535 caractères.
memory_usage	100	Multiplicateur d'allocation dynamique de mémoire. Ajustez cette valeur vers le haut ou vers le bas pour réguler l'utilisation de la mémoire totale du serveur.
temp_directory	"	Indique le répertoire à utiliser pour les fichiers temporaires.
request_passwords	Y	Contrôle si vous êtes invité à entrer un mot de passe de base de données lorsque vous exécutez un flux publié exigeant un accès à une base de données (Y (Oui)/N (Non)).

Utilisation du fichier de paramètres pour personnaliser l'exécution de flux

Lorsqu'un flux est publié, le fichier image contient des informations sur la source de données et la destination des données de sortie sélectionnées dans le flux d'origine. Toutefois, il est souvent pratique d'exécuter un flux en fonction de données issues d'une autre source de données ou de rediriger les données de sortie vers une autre destination. Vous devez, pour cela, modifier le fichier de paramètres du flux.

Cette opération s'avère particulièrement importante lorsque vous exécutez le flux publié sur un autre ordinateur que celui de sa création. Vous devrez mettre à jour les emplacements des fichiers d'entrée et de sortie pour refléter la structure de l'ordinateur cible. Les flux publiés ne prenant pas en charge les chemins relatifs, il est extrêmement important de vérifier l'emplacement des données de votre fichier de paramètres. (Si vous indiquez des chemins relatifs dans les options du noeud d'exportation lors de la publication du flux, IBM SPSS Modeler les convertit en chemins absolus avant de créer le fichier image et le fichier de paramètres du flux publié.)

Vous pouvez également utiliser le fichier des paramètres pour définir les options d'exécution et modifier les valeurs des paramètres de flux.

Les valeurs de paramètre sont spécifiées dans le fichier de paramètres sous forme de paires <name.attribute>=<value>. Le nom de certains paramètres contient un numéro d'identification permettant de distinguer les noeuds appartenant au même type dans le flux. Par exemple, dans un flux où l'entrée et la sortie font référence à une connexion de base de données, les paramètres décrivant la connexion d'entrée peuvent être dbconn0.datasourc, dbconn0.user, etc., et les paramètres associés à la connexion de sortie, dbconn1.datasourc, dbconn1.user, etc. Les valeurs sont systématiquement placées entre guillemets doubles. Le fichier de paramètres contient les paramètres suivants qui peuvent tous être modifiés pour personnaliser l'exécution du flux.

Tableau 3. Paramètres d'exécution.

Paramètre	Rôle
dbconnx.datasourc	Nom de source de données (DSN).
dbconnx.user	Nom d'utilisateur destiné aux bases de données à accès limité.
dbconnx.password	Mot de passe destiné aux bases de données à accès limité.
dbconnx.epassword	Mot de passe codé destiné aux bases de données à accès limité. Pour créer un mot de passe codé, sélectionnez Coder le mot de passe dans le menu Outils de l'interface utilisateur de IBM SPSS Modeler. Copiez et collez le mot de passe codé selon vos besoins.
dbconnx.password_required	Indicateur indiquant si un mot de passe doit être entré pour la connexion à cette base de données. (La valeur est Y (Oui) ou N (Non).) Ce paramètre adopte automatiquement la valeur N si aucun mot de passe n'est spécifié lors de la publication.
dbconnx.catalog	Catalogue associé à la connexion de base de données.
dbtablex.name	Nom du tableau de base. <i>Remarque</i> : L'ordre des champs dans un nouveau tableau doit être identique à celui des champs du premier flux publié.
dbtablex.schema	Nom du schéma.
dbtablex.catalog	Catalogue associé au tableau actuel.
filex.name	Nom du fichier de données. <i>Remarque</i> : L'ordre des champs dans un nouveau fichier d'entrée doit être identique à celui des champs du premier flux publié.
filex.path	Emplacement du fichier (nom du répertoire).

Tableau 3. Paramètres d'exécution (suite).

Paramètre	Rôle
filex.field_names_included	En ce qui concerne les fichiers texte, il s'agit de l'indicateur indiquant si le nom des champs apparaît sur la première ligne du fichier. (La valeur est Y (Oui) ou N (Non).)
filex.field_separator	Caractères permettant de séparer les valeurs des champs des fichiers texte.
filex.decimal_separator	Séparateur décimal des fichiers texte.
options.angle_in_radians	Indicateur indiquant si des radians servent d'unité de mesure dans les expressions trigonométriques CLEM. (La valeur est Y (Oui) ou N (Non).)
options.date_2digit_baseline	Définit le siècle des dates dont l'année porte sur deux chiffres.
options.date_baseline	La date de référence (toujours le 1er janvier) utilisée par les fonctions de date CLEM opérant avec une date unique.
options.time_rollover	Indicateur indiquant si les différences horaires négatives font référence à un moment passé.
options.decimal_separator	Séparateur décimal par défaut des fichiers texte.
options.time_format	Format horaire utilisé lorsque les chaînes sont interprétées comme des heures par les fonctions horaires de CLEM.
options.date_format	Format de date utilisé lorsque les chaînes sont interprétées comme des dates par les fonctions de date de CLEM.
options.timestamp_format	Format utilisé lorsque les champs d'horodatage sont lus comme des chaînes depuis les sources de données ODBC.
paramx.value	Valeur des paramètres de flux. <i>Remarque</i> : Le nom, la description et le type de stockage des paramètres sont inclus comme référence, mais ils ne peuvent pas être modifiés. La valeur du type doit être une valeur valide.

Intégration de IBM SPSS Modeler Solution Publisher dans les applications

Outre le moteur autonome Runtime, IBM SPSS Modeler Solution Publisher propose une bibliothèque de programmation (CLEMRTL) qui permet à d'autres programmes de contrôler l'exécution des flux IBM SPSS Modeler Solution Publisher. Vous pouvez appeler des procédures CLEMRTL dans les programmes client écrits en C et en C++. Pour utiliser la bibliothèque CLEMRTL, vous devez insérer le fichier d'en-tête *clemrtl.h* (disponible dans le dossier *..\installation\clemrtl\include*) et veiller à lier le fichier de bibliothèque approprié à la plate-forme de développement dans votre application lors de sa création. Le fichier *clemrtl.dll* est disponible dans le dossier *..\installation\bin*.

Tout fichier source faisant référence aux procédures de bibliothèque doit intégrer le fichier d'en-tête *clemrtl.h*. Ce fichier fournit des prototypes ANSI C concernant les procédures de bibliothèque et désigne les macros utiles. Les seuls en-têtes nécessaires sont ceux exigés par le programme. Pour éviter tout conflit de dénomination, les noms des types de bibliothèque et des fonctions commencent par *clemrtl_* et ceux des macros par *CLEMRTL_*.

La bibliothèque CLEMRTL génère des rapports contenant des informations utiles qui doivent être renvoyées à l'application d'une manière ou d'une autre. CLEMRTL offre trois mécanismes de traitement de ces messages :

- Les messages peuvent être inscrits dans un fichier journal spécifié via la fonction `clemrtl_setLogFile()`.

- L'application peut récupérer des détails de la dernière erreur à l'aide de la fonction `clemrtl_getErrorDetail()`.
- Elle propose sa propre procédure de gestion des rapports à l'aide de la fonction `clemrtl_setReportHandler()`.

Le système Runtime possède un catalogue de messages traduits destinés aux rapports. Tout rapport transmis à l'application inclut la chaîne du message traduit. Les rapports intègrent également le code du rapport afin que l'application puisse décider d'interpréter et de présenter le message différemment. Pour pouvoir obtenir des messages détaillés (traduits), l'application doit placer le fichier `messages.cfg` approprié dans un répertoire `config`. Différentes versions du fichier `messages.cfg` correspondant aux paramètres régionaux se trouvent dans le répertoire `<répertoire d'installation>/config/<paramètres régionaux>`, où `<répertoire d'installation>` est le répertoire dans lequel vous avez installé IBM SPSS Modeler Solution Publisher et `<paramètres régionaux>` est le paramètre régional choisi. Sélectionnez la version correcte du fichier `messages.cfg` et ajoutez-la à un sous-répertoire `config` du répertoire des projets.

Le système Runtime ne nécessite pas d'environnement ni de paramètres de registre particuliers pour pouvoir fonctionner correctement. Des bibliothèques dépendantes doivent être distribuées avec une application liée à CLEMRTL. Ces fichiers sont inclus dans le répertoire `<répertoire d'installation>/bin`.

Configuration requise

Le tableau ci-après contient les plateformes et les compilateurs testés avec la bibliothèque de programmation Runtime.

Tableau 4. Plateformes et compilateurs.

Plateforme	Compilateur(s) suggéré(s)
AIX	IBM XL C/C++ Enterprise Edition pour AIX version 8.0
Solaris 64 bits	Sun Studio 11 5.8 correctif 121017-06
Linux 64 bits	gcc 3.2.3, Intel C++ Compiler 9.1.043
fenêtres	Visual Studio 2005

Exemple d'application IBM SPSS Modeler Solution Publisher

L'application fournie à titre d'exemple, `dlltest.c`, est incluse dans le dossier `\clemtrl\demo` du répertoire d'installation de IBM SPSS Modeler Solution Publisher. Utilisez cette application pour vous initier à l'intégration de la bibliothèque Runtime dans vos propres programmes.

Les informations utilisées par les applications fournies à titre d'exemple sont incluses dans le dossier `\clemtrl\demo`. Reportez-vous au fichier `README.txt` pour plus d'instructions. La compilation d'une application variant en fonction de la plate-forme, des fichiers exemple sont également fournis.

Chapitre 4. Installation d'IBM SPSS Modeler Solution Publisher Runtime

Installation de Runtime

L'installation d'IBM SPSS Modeler Solution Publisher Runtime vous permet d'exécuter des flux publiés. Les flux publiés sont indépendants du système d'exploitation ; vous pouvez donc exécuter tout flux publié sur n'importe quelle plate-forme prise en charge par Runtime.

La fonctionnalité qui vous permet de publier des flux depuis IBM SPSS Modeler est installée avec IBM SPSS Modeler. Notez que les fichiers publiés doivent être réexportés avec chaque version d'IBM SPSS Modeler. Par exemple, un fichier exporté à l'aide d'une version antérieure ne fonctionnera pas avec la version 17 d'IBM SPSS Modeler Solution Publisher Runtime.

Configuration requise

La configuration système requise pour l'installation de IBM SPSS Modeler Solution Publisher Runtime est la suivante :

- **Système d'exploitation.** Windows Server 2008 Standard Edition pour les systèmes 32 bits x86 ; Windows Server 2008 Enterprise Edition pour les systèmes 64 bits x64 ; Windows Server 2008 Enterprise Edition R2 pour les systèmes 64 bits x64 ; Windows Server 2003 Standard Edition R2 pour les systèmes 32 bits x86 ou 64 bits x64 IBM AIX[®] 6.1 ou 7.1 pour les systèmes POWER 64 bits ; Oracle Solaris[™] 9.x ou 10 pour les systèmes 64 bits SPARC ; Red Hat Enterprise Linux 5.x pour les systèmes 32 bits x86 ; Red Hat Enterprise Linux 5.x ou 6.x pour les systèmes 64 bits x64 ou IBM System z ; Red Hat Enterprise Linux Advanced Platform 5.x pour les systèmes 32 bits x86 ; Red Hat Enterprise Linux Advanced Platform 5.x ou 6.x pour les systèmes 64 bits x64 ou IBM System z ; SuSE Linux Enterprise Server 10 ou 11 pour les systèmes 64 bits x64 ou IBM System z
- **Matériel.** Processeur Pentium[®] ou de type Pentium ou version ultérieure (pour Windows 32 bits) ; gamme de processeurs x64 (AMD 64 et EM64T) (pour Windows 64 bits) s'exécutant à 1 GHz ou plus rapide Processeur PowerPC, 233 MHz ou plus rapide et IBM System p pour IBM AIX ; gamme de processeurs UltraSPARC II (ou plus) pour Solaris ; processeur Pentium ou de type Pentium ou version ultérieure pour Linux 32 bits ; gamme de processeurs x64 (AMD 64 et EM64T) ou IBM s390x System z pour Linux 64 bits
- **Espace disque libre minimum.** 2 gigaoctets (Go) recommandés.
- **Mémoire RAM.** 4 Go recommandés.
- **Environnement virtuel.** Les environnements virtuels suivants prennent en charge IBM SPSS Modeler Server.
 - IBM PowerVM Hypervisor pour IBM AIX
 - IBM PR/SM ou z/VM pour Linus pour IBM System z
 - VMWare ESX Server 4.1
 - VMWare vSphere 4.0
- **Environnement de cluster.** Les environnements de cluster suivants prennent en charge IBM SPSS Modeler Server.
 - IBM HACMP/RSCT pour IBM AIX
 - Serveur d'applications IBM WebSphere 64 bits
 - Microsoft Cluster Service pour Windows
 - Oracle Solaris Cluster
 - Serveur d'applications Oracle WebLogic 64 bits
 - Red Hat Cluster Suite pour Red Hat Enterprise Linux

Installation sous Windows

La section suivante décrit la procédure d'installation de SPSS Modeler Solution Publisher Runtime sous le système d'exploitation Windows.

1. Insérez le disque d'installation d'IBM SPSS Modeler Solution Publisher dans le lecteur ou accédez au support d'installation par d'autres moyens.
2. Dans le menu d'installation, choisissez **Install IBM SPSS Modeler Solution Publisher Runtime** ou **Install IBM SPSS Modeler Solution Publisher Runtime for Win64**, en fonction du système sur lequel vous choisissez d'effectuer l'installation.
3. Dans l'assistant d'installation, cliquez sur **Suivant** pour commencer.
4. Suivez les instructions qui s'affichent à l'écran. Pour continuer, cliquez sur **Suivant**.
5. Une fois que vous avez défini toutes les options, vous êtes en mesure de lancer l'installation. Cliquez sur l'option d'**installation** pour commencer le transfert des fichiers.
6. Une fois tous les fichiers installés, cliquez sur **Terminer**.

Installation sous UNIX

Les sections suivantes décrivent la configuration système supplémentaire requise pour l'installation de IBM SPSS Modeler Solution Publisher Runtime sous le système d'exploitation UNIX, et la procédure à suivre.

Éléments de configuration supplémentaires pour UNIX

Vous devez vous assurer que les limites du noyau sur le système sont suffisantes pour utiliser IBM SPSS Modeler Server. Les limites des données, de la mémoire et les valeurs ulimit du fichier sont particulièrement importantes et doivent être définies sur illimitées au sein de l'environnement IBM SPSS Modeler Server. Pour ce faire :

1. Ajoutez les commandes suivantes à *modelersrv.sh* :

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

En outre, définissez la limite maximale de pile autorisée par le système (ulimit -s XXXX), par exemple :

```
ulimit -s 64000
```

2. Redémarrez IBM SPSS Modeler Server.

Vous avez également besoin de l'utilitaire de compression de fichiers *gzip* et de *GNU cpio* installés au bon emplacement afin que le programme d'installation puisse décompresser les fichiers d'installation. En outre, sur la machine qui exécute SPSS Modeler Server, vous devez définir l'environnement local sur EN_US.UTF-8.

Procédure d'installation sous UNIX

Ces instructions d'installation s'appliquent à IBM SPSS Modeler Solution Publisher Runtime pour UNIX. Vous devez disposer d'un droit d'accès en lecture et en écriture sur le répertoire d'installation cible ; par conséquent, connectez-vous via un compte qui présente un niveau d'autorisation suffisant.

Remarque : Vous devez vous connecter comme utilisateur *root* pour effectuer l'installation.

1. Insérez le disque d'installation dans le lecteur et assurez-vous que vous pouvez accéder au lecteur. Sous Solaris, le gestionnaire de volume (démon vold) doit être en cours d'exécution.
2. Remplacez les répertoires par le point de montage du DVD.
3. Remplacez les répertoires par le répertoire *modelrun*.
4. Modifiez pour le répertoire de la plateforme approprié.
5. Exécutez le script d'installation *.bin*. Par exemple :

- ```
./modelersolutionpublisherhpa64.bin -i console
```
6. Les détails de l'introduction apparaissent. Appuyez sur Entrée pour continuer.
  7. Les informations de licence apparaissent. Lisez la licence, saisissez 1 pour l'accepter puis appuyez sur Entrée pour continuer.
  8. Une liste des langues disponibles apparaît. Saisissez le numéro de la langue à installer et appuyez sur Entrée pour continuer.
  9. Une invite vous demande de saisir l'emplacement de l'installation. Pour utiliser le répertoire par défaut `/usr/IBM/SPSS/ModelerSolutionPublisher<nn>` (où `<nn>` est le numéro de version), appuyez sur Entrée. Le programme Runtime sera installé dans le répertoire indiqué.
  10. Une invite vous demande de confirmer l'emplacement de l'installation. Lorsque celui-ci est correct, tapez `y` et appuyez sur Entrée.
  11. Un récapitulatif de préinstallation apparaît pour confirmer vos entrées. Appuyez sur Entrée pour continuer.
  12. Un message apparaît pour vous informer que la routine d'installation est prête à être exécutée. Appuyez sur Entrée pour continuer.
  13. Une barre de progression apparaît pendant l'exécution de l'installation. Lorsque l'installation est terminée, appuyez sur Entrée pour quitter le programme d'installation.

## Configuration d'ODBC sous UNIX

Par défaut, le gestionnaire de pilote DataDirect n'est pas configuré pour qu'IBM SPSS Modeler Solution Publisher Runtime utilise ODBC sur les systèmes UNIX. Pour configurer UNIX afin qu'il charge le gestionnaire de pilote DataDirect, saisissez les commandes suivantes (où `rép_installation_sp` est le répertoire d'installation de Solution Publisher Runtime) :

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

## Dépannage d'une installation

### Signature numérique non valide lors de l'installation

Les produits IBM SPSS Modeler utilisent la certification IBM pour la signature numérique. Dans certaines circonstances, vous pouvez obtenir l'erreur suivante lors de la tentative d'installation des produits SPSS Modeler :

Erreur 1330. Impossible d'installer un fichier requis car le nom du fichier CAB a une signature numérique non valide...

### Tous les utilisateurs Windows

Ce message s'affiche lorsque vous tentez d'installer les produits SPSS Modeler sur une machine qui ne dispose pas d'une connexion Internet et sur laquelle le certificat correct n'est pas installé. Suivez la procédure ci-dessous pour résoudre ce problème.

1. Cliquez sur **OK** à l'apparition du message.
2. Cliquez sur **Annuler** pour quitter le programme d'installation.
3. Si l'ordinateur sur lequel vous souhaitez effectuer l'installation n'a pas de connexion Internet, effectuez l'étape suivante sur un ordinateur avec une connexion Internet et copiez le fichier `.cer` sur l'ordinateur sur lequel vous souhaitez effectuer l'installation.
4. Depuis <http://www.verisign.com/support/roots.html>, défilez vers le bas pour rechercher l'autorité de certification (CA) racine **VeriSign Class 3 Primary CA - G5**. Téléchargez cette CA racine et enregistrez-la en tant que fichier `.cer`.
5. Faites un double clic sur le fichier `.cer`.
6. Dans l'onglet Général, cliquez sur **Installer le certificat**.

7. Suivez les instructions dans l'assistant d'importation de certificat, à l'aide des options par défaut et cliquez sur **Terminer** pour finir.
8. Effectuez de nouveau l'installation.

---

## Démarrage du programme Runtime

Une fois que vous avez installé le moteur d'exécution, vous pouvez l'utiliser pour exécuter des flux qui ont été publiés à partir de IBM SPSS Modeler à l'aide des noeuds d'exportation. Pour plus d'informations, voir la rubrique «Publication de flux», à la page 8. Lancez Runtime à partir de la ligne de commande, en entrant des options indiquant le nom du fichier image publié et un fichier de paramètres facultatif à exécuter.

Pour exécuter Runtime, saisissez ce qui suit à l'invite de commande :

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

Les entrées de la commande sont répertoriées dans le tableau ci-après.

Tableau 5. Entrées de commande Runtime

| Entrée de commande | Description                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <parameter-file>   | correspond au nom du fichier de paramètres publié (facultatif).                                                                                                                                  |
| <options>          | est une liste séparée par des virgules de paires option=vaue précisant les options d'exécution du programme Runtime. <b>Remarque</b> : Il ne doit pas y avoir d'espaces vides après une virgule. |
| <image file>       | est le nom du fichier image publié (*.pim) à exécuter.                                                                                                                                           |

*Remarque* : Si vous utilisez des langues multioctet comme le chinois ou le japonais dans un environnement UNIX, vous devez spécifier les codes de langue à utiliser. L'exemple suivant présente les codes de langue du chinois simplifié :

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

où -p est l'emplacement des fichiers pim et par.

Consultez Chapitre 3, «Exécution de flux publiés», à la page 11 pour plus d'informations.

## Changement de répertoire temporaire

Certaines opérations effectuées par IBM SPSS Modeler Solution Publisher Runtime peuvent nécessiter la création de fichiers temporaires. Par défaut, Runtime crée les fichiers temporaires dans le répertoire temporaire du système. Vous pouvez modifier l'emplacement du répertoire temporaire en effectuant les opérations suivantes.

*Remarque* : Il est très important d'utiliser la syntaxe correcte et de prêter une attention particulière aux espaces et au chemin du répertoire.

Supposons que vous utilisiez IBM SPSS Modeler Solution Publisher Runtime avec les fichiers publiés *webtest.pim* et *webtest.par*. Vous souhaitez modifier l'emplacement des fichiers temporaires et utiliser le répertoire *C:\published\temp*. Pour modifier l'emplacement du répertoire temporaire utilisé par l'exécutable :

1. Dans le répertoire où les fichiers *.pim* et *.par* sont enregistrés, exécutez la commande suivante :

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

Cette commande indique à Runtime d'utiliser "*C:\published\temp*" comme répertoire temporaire pour ce processus particulier uniquement.

---

## Désinstallation de IBM SPSS Modeler Solution Publisher Runtime

La section suivante décrit les procédures de désinstallation de IBM SPSS Modeler Solution Publisher Runtime.

### Procédure de désinstallation sous Windows

1. Dans le Panneau de configuration de Windows, sélectionnez **Ajout/Suppression de programmes**.
2. Dans la liste, sélectionnez **IBM SPSS Modeler Solution Publisher <nn>**.
3. Cliquez sur **Supprimer**.
4. Suivez ensuite les instructions à l'écran pour supprimer le programme Runtime.

*Remarque* : Vous devez être connecté à votre ordinateur en tant qu'administrateur pour supprimer des programmes.

### Procédure de désinstallation sous UNIX

Supprimez le répertoire d'installation, y compris tous les fichiers installés.



---

## Chapitre 5. API IBM SPSS Embedded Predictive Modeling

L'API IBM SPSS Embedded Predictive Modeling définit une interface Java simple permettant d'intégrer les artefacts conçus IBM SPSS Modeler dans vos applications d'une manière incorporée. La documentation complète dans le format Javadoc est incluse dans IBM SPSS Modeler Solution Publisher pour les programmeurs qui utilisent la bibliothèque.

Notez les restrictions suivantes lors de l'utilisation de l'API Embedded Predictive Modeling (EPM) :

- **Des libellés de noeud source et terminal uniques sont nécessaires.** L'API EPM utilise des libellés de noeud. Bien que les libellés soient toujours dans un contexte de noeud source ou terminal, ils peuvent être ambigus.
- **Les super noeuds ne sont pas pris en charge comme noeuds terminaux.** Pour fournir une source commune pour les super noeuds dans toutes les opérations, ils ne sont pas pris en charge comme noeuds source ou terminaux.
- **Restrictions dans les noms de zone d'entrée.** Certains caractères ne peuvent pas être utilisés dans les noms de zone.

Voir les informations suivantes pour une présentation de la fonctionnalité de l'API. Pour plus d'informations, voir la documentation Javadoc fournie avec le produit.

### Evaluation incorporée de modèle prédictif

**Usage :** utilisée par l'application pour générer une analyse prédictive à la demande à mesure que de nouvelles données entrent. L'application doit gérer des modèles actualisés. Les noeuds source, et le noeud terminal de la branche d'évaluation, sont remplacés par les entrées et sorties de l'application dans ce mode d'évaluation.

**Action :** prépare la branche d'évaluation que vous avez créée dans le flux IBM SPSS Modeler une fois pour les appels répétitifs qui génèrent rapidement et efficacement l'analyse prédictive depuis les données d'entrée gérées par l'application.

### Actualisation de modèle prédictif

**Usage :** exécutez l'apprentissage initial des algorithmes du modèle dans la conception de flux en utilisant les données historiques. En outre, "actualisez" régulièrement les algorithmes de modèle prédictif configurés avec les données historiques les plus actuelles.

**Action :** exécute tous les noeuds de génération de modèle prédictif dans le flux IBM SPSS Modeler, ce qui permet d'effectuer l'apprentissage de l'algorithme du modèle prédictif en utilisant des techniques d'exploration. Met à jour les nuggets d'applicateur de modèle liés à chaque noeud de création de modèle. Les paramètres de flux et les propriétés de noeud source/terminal peuvent être utilisés pour implémenter un certain niveau de dynamisme.

### Evaluation de modèle prédictif

**Usage :** utilisée par une application pour communiquer l'exactitude, la confiance ou une autre évaluation qualitative d'un modèle prédictif formé dans le flux IBM SPSS Modeler. Ces informations sont généralement utilisées pour contrôler la promotion d'un modèle prédictif nouvellement actualisé dans l'évaluation dans l'application. L'actualisation du modèle retourne un nouvel objet `PredictiveModel` principalement pour permettre la comparaison des données d'évaluation du modèle nouvellement actualisé aux données d'origine.

**Action** : exécute tous les noeuds de création de document (principalement les noeuds de sortie non interactifs des palettes de graphique et de sortie IBM SPSS Modeler) dans le flux IBM SPSS Modeler qui retourne les documents résultants. Les paramètres de flux et les propriétés de noeud source/terminal peuvent être utilisés pour implémenter un certain niveau de dynamisme.

## **Exécution du flux du modèle prédictif**

**Usage** : tous les scénarios d'utilisation qui ne sont pas adaptés à "l'actualisation", l'"évaluation" ou aux deux modes d'évaluation doivent être traités en utilisant l'exécution de flux.

**Action** : exécution contrôlée par les options de flux IBM SPSS Modeler. Soit dans le mode "Tous les noeuds terminaux" qui revient à cliquer sur la grande flèche verte dans IBM SPSS Modeler Client, ou tel que cod dans le script au niveau flux. Les paramètres de flux et les propriétés de noeud source/terminal peuvent être utilisés pour implémenter un certain niveau de dynamisme.

## **Données externes d'évaluation du modèle prédictif**

**Usage** : appelées processus "par lot" du fait de la manière dont un lot complet de données d'entrée est traité.

**Action** : exécute la branche d'évaluation conformément à la conception. Les entrées des noeuds source de données configurés et d'analyse prédictive générées persistent, tel que contrôlé par le noeud terminal. Les paramètres de flux et les propriétés de noeud source/terminal peuvent être utilisés pour implémenter un certain niveau de dynamisme.

## **Paramètres de flux de modèle prédictif et spécifications des propriétés des noeuds source et terminaux**

**Usage** : une méthode courante consiste à définir les paramètres référencés dans les expressions dans les branches du flux. Un autre point de contrôle consiste à modifier certaines propriétés d'un noeud source ou terminal pour modifier le chemin de fichier utilisé, etc.

**Action** : certains aspects de l'exécution du flux IBM SPSS Modeler peuvent être modifiés dynamiquement par une application.

**Remarque** : Les modifications de paramètres doivent être effectuées avant de créer le PredictiveScorer (voir la description d'évaluation incorporée ci-dessus), et les propriétés des noeuds source et terminaux n'ont aucune signification pour l'objet.

---

# Chapitre 6. Référence de l'API de la bibliothèque d'exécution IBM SPSS Modeler Solution Publisher

---

## Introduction

L'API CLEMRTL permet de contrôler l'exécution des flux publiés à partir de votre application. Cette section décrit les fonctions API disponibles.

Les points suivants se rapportent à l'utilisation générale de l'API :

- L'API comporte une liaison C qui offre une compatibilité maximale ; la bibliothèque présente, quant à elle, des dépendances C++. De ce fait, elle ne peut être utilisée qu'avec un éditeur de liens C++ sur certaines plates-formes.
- Vous devez appeler la fonction `clemrtl_initialize_ext()` avant d'utiliser toute autre fonction dans la bibliothèque.
- Le type `clemrtl_image_handle` est utilisé comme identificateur d'image.
- Chaque fonction renvoie un code indicateur de statut. Les valeurs de l'indicateur de statut sont présentées dans le tableau ci-après.

Tableau 6. Valeurs d'indicateur de statut

| Résultat      | Description                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|
| CLEMRTL_OK    | Succès                                                                                                                             |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires                                                                                        |
| CLEMRTL_ERROR | Echec avec informations supplémentaires sur l'erreur disponibles par le biais de la fonction <code>clemrtl_getErrorDetail()</code> |

## Présentation du processus d'API

Une application qui utilise l'API obéit, dans les grandes lignes, au schéma suivant :

1. Initialisez la bibliothèque à l'aide de `clemrtl_initialize_ext()`.
2. Vérifiez et modifiez les options d'exécution en utilisant `clemrtl_getOption()` et `clemrtl_setOption()`.
3. Ouvrez une image à l'aide de `clemrtl_openImage()` et recevez un descripteur d'image.
4. Vérifiez et modifiez les paramètres d'image en utilisant `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` et `clemrtl_setParameter()`.
5. Exécutez l'image à l'aide de `clemrtl_execute()`. Si la même image doit être exécutée plusieurs fois sans modification des paramètres, utilisez d'abord `clemrtl_prepare()`.
6. Fermez l'image à l'aide de `clemrtl_closeImage()`.
7. Pour annuler une exécution en cours, utilisez `clemrtl_interrupt()`.
8. Pour extraire les informations concernant la dernière erreur, utilisez `clemrtl_getErrorDetail()`. Pour recevoir des messages d'erreur, ainsi que d'autres messages de diagnostic, au fur et à mesure de leur génération, exécutez la fonction `clemrtl_setReportHandler()`.

La bibliothèque est conçue pour une utilisation avec plusieurs unités d'exécution sous les conditions suivantes :

- Les fonctions globales (celles n'utilisant pas de descripteur d'image comme argument) ne peuvent pas être appelées simultanément.

- Les fonctions propres aux images (celles utilisant un descripteur d'image comme argument) ne peuvent pas être appelées simultanément lorsqu'elles s'appliquent au même descripteur d'image, à l'exception de `clemrtl_interrupt()`.
- Les fonctions locales d'image peuvent être appelées simultanément lorsqu'elles s'appliquent à des descripteurs d'image différents.

Plus particulièrement, cela signifie que des unités d'exécution distinctes peuvent préparer et exécuter des images simultanément, à condition d'utiliser des descripteurs d'image différents. Des descripteurs d'image différents peuvent faire référence à la même image (la fonction `clemrtl_openImage()` crée un descripteur chaque fois qu'elle est appelée, et ce également pour la même image) mais, lors de l'exécution de plusieurs instances d'une même image, veillez à modifier les paramètres d'image pour rediriger la sortie comme souhaité. Si deux images exécutées simultanément tentent d'écrire dans le même fichier de sortie ou dans la même table de base de données, les résultats seront imprévisibles.

## Fonctions API

Retrouvez ci-dessous la liste complète des fonctions présentées dans l'API.

### initialise

Cette fonction a été remplacée par

```
initialise_ext(indicateurs, 0, 0);
```

Les nouveaux programmes doivent appeler *initialise\_ext*.

### initialise\_ext

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

Initialise IBM SPSS Modeler Solution Publisher Runtime. *Cette fonction doit être appelée avant toute autre fonction API.*

Tableau 7. Paramètres *initialise\_ext*

| Paramètre | Description                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flags     | Contrôle certains aspects du processus d'initialisation. La valeur est construite en tant qu'opérateur OR sur les bits des indicateurs décrits ci-dessous.                                          |
| arg_count | Le nombre d'arguments d'initialisation supplémentaires. Ce nombre doit être inférieur ou égal à la longueur de l'ensemble de args.                                                                  |
| args      | Arguments d'initialisation supplémentaires. Les arguments sont décrits par type :<br><pre>typedef struct _clemrtl_init_arg {     const char* nom;     const char* valeur; } clemrtl_init_arg;</pre> |

Le champ `nom` est le nom d'une propriété de configuration et le champ `valeur` est sa valeur. Les propriétés de configuration indiquées dans le tableau ci-après sont reconnues.

Tableau 8. Propriétés de configuration

| Nom                                 | Valeur                                                                                                                                                                               |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>installation_directory</code> | Le chemin d'accès entier au dossier d'installation IBM SPSS Modeler Solution Publisher Le dossier d'installation qui contient les dossiers <code>bin</code> et <code>config</code> . |



Renvoie l'un des codes d'erreur affichés dans le tableau ci-après.

Tableau 9. Codes d'erreur

| Résultat     | Description                                                   |
|--------------|---------------------------------------------------------------|
| CLEMRTL_OK   | Réussite.                                                     |
| CLEMRTL_FAIL | Echec d'initialisation et absence de détails supplémentaires. |

Le paramètre `flags` permet de contrôler certains aspects du processus d'initialisation. La valeur 0 (ou `CLEMRTL_INIT_DEFAULTS`) définit le comportement par défaut suivant :

- Définissez le fuseau horaire par défaut en appelant `tzset()`.
- Affecte la valeur "C" au paramètre régional `LC_NUMERIC`.
- Définissez le nouveau gestionnaire global de manière à ce qu'il génère une exception quand la mémoire est épuisée.

Le comportement peut être modifié en définissant des combinaisons des indicateurs répertoriés dans le tableau ci-après.

Tableau 10. Indicateurs

| Indicateur                                  | Description                                                      |
|---------------------------------------------|------------------------------------------------------------------|
| <code>CLEMRTL_INIT_NO_TZ</code>             | Ne pas définir le fuseau horaire.                                |
| <code>CLEMRTL_INIT_NO_LOCALE</code>         | Ne pas modifier les paramètres régionaux.                        |
| <code>CLEMRTL_INIT_NO_NEW_HANDLER</code>    | Ne pas paramétrer le nouveau gestionnaire.                       |
| <code>CLEMRTL_INIT_LOCAL_NEW_HANDLER</code> | Localiser le nouveau gestionnaire au sein de chaque appel d'API. |

La bibliothèque exige que la valeur "C" soit affectée au paramètre régional `LC_NUMERIC` pour pouvoir réaliser une conversion correcte entre les nombres et les chaînes (le format de nombre IBM SPSS Modeler utilise les règles du paramètre régional "C"). Cela concerne également la conversion des nombres lus et écrits dans les fichiers texte.

- Indiquez `NO_LOCALE` si l'application dépend du paramètre régional `LC_NUMERIC` et n'exige pas de conversion entre nombres et chaînes.
- `NO_NEW_HANDLER` et `LOCAL_NEW_HANDLER` s'excluent mutuellement. La bibliothèque exige que la valeur `new` génère une exception lorsque la mémoire est épuisée.
- Indiquez `NO_NEW_HANDLER` si l'application définit elle-même son nouveau gestionnaire ou si vous êtes certain que le compilateur/système d'exécution génère une exception par défaut.

Spécifiez `LOCAL_NEW_HANDLER` seulement si l'application suppose que la valeur `new` renvoie un pointeur nul lorsque la mémoire est épuisée. *Remarque* : L'utilisation de ce commutateur n'est pas recommandée dans un environnement multi-unité d'exécution.

## getOption

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

Extrait la valeur d'une option d'exécution.

Tableau 11. Paramètres `getOption`

| Paramètre         | Description      |
|-------------------|------------------|
| <code>name</code> | Nom de l'option. |

Tableau 11. Paramètres `getOption` (suite)

| Paramètre               | Description                              |
|-------------------------|------------------------------------------|
| <code>value</code>      | Tampon qui reçoit la valeur de l'option. |
| <code>value_size</code> | Taille du tampon de valeurs.             |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 12. Codes de statut

| Résultat                   | Description                                   |
|----------------------------|-----------------------------------------------|
| <code>CLEMRTL_OK</code>    | Réussite.                                     |
| <code>CLEMRTL_FAIL</code>  | Echec et absence de détails supplémentaires.  |
| <code>CLEMRTL_ERROR</code> | Echec et présence de détails supplémentaires. |

L'opération échoue si le nom de l'option ne correspond pas à l'un des noms d'option d'exécution répertoriés ci-dessus (par exemple, `memory_usage` ou si la taille du tampon de valeurs n'est pas suffisante pour recevoir la valeur accompagnée d'un caractère de fin nul. Les valeurs d'option sont toujours des chaînes, même dans le cas d'une interprétation numérique. Consultez Chapitre 3, «Exécution de flux publiés», à la page 11 pour plus d'informations.

## setOption

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

Définit la valeur d'une option d'exécution.

Tableau 13. Paramètre `setOption`

| Paramètre          | Description         |
|--------------------|---------------------|
| <code>name</code>  | Nom de l'option.    |
| <code>value</code> | Valeur de l'option. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 14. Codes de statut

| Résultat                   | Description                                   |
|----------------------------|-----------------------------------------------|
| <code>CLEMRTL_OK</code>    | Réussite.                                     |
| <code>CLEMRTL_FAIL</code>  | Echec et absence de détails supplémentaires.  |
| <code>CLEMRTL_ERROR</code> | Echec et présence de détails supplémentaires. |

L'opération échoue si le nom de l'option ne correspond pas à l'un des noms d'option d'exécution répertoriés ci-dessus (par exemple, `memory_usage`). Les valeurs d'option sont toujours des chaînes, même dans le cas d'une interprétation numérique.

## setLogFile

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

Redirige les messages de consignation à partir de toutes les images ouvertes ultérieurement.

Tableau 15. Paramètres setLogFile

| Paramètre | Description                                                                                                                                                                                                                                                    |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mode      | Le mode de consignation doit présenter l'une des valeurs suivantes :<br>CLEMRTL_NULL_LOG—suppress messages.<br>CLEMRTL_STDERROR_LOG—send messages to the standard error output.<br>CLEMRTL_FILE_LOG—send messages to the file named by the log_file parameter. |
| log_file  | Chaîne contenant le nom du fichier.                                                                                                                                                                                                                            |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 16. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si le mode ne correspond pas à l'un des modes reconnus, ou si une consignation dans un fichier est nécessaire et que le nom de fichier n'est pas spécifié ou n'est pas valide.

## openImage

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

Ouvre une image de flux publié.

Tableau 17. Paramètres openImage

| Paramètre  | Description                       |
|------------|-----------------------------------|
| image_file | Nom du fichier image.             |
| param_file | Nom du fichier de paramètres.     |
| handle     | Reçoit le descripteur de l'image. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 18. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

Le fichier de paramètres peut être NULL si aucun fichier de paramètres distinct n'est nécessaire. L'opération échoue si le fichier image n'est pas spécifié, si l'un des deux noms de fichier n'est pas valide ou si le contenu est illisible. Si l'opération réussit, le descripteur d'image renvoyé peut être utilisé pour identifier l'instance d'image dans les appels d'API suivants.

## closeImage

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

Ferme un descripteur d'image et libère ses ressources.

Tableau 19. Paramètres closeImage

| Paramètre | Description             |
|-----------|-------------------------|
| handle    | Descripteur de l'image. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 20. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si un autre appel d'API est en cours sur l'image, particulièrement si cette dernière est toujours en cours d'exécution. Si l'opération réussit, l'image est fermée et le descripteur ne peut plus être utilisé dans aucun appel d'API suivant, sauf s'il s'agit d'un autre appel de `clemrtl_closeImage()` (ce qui n'a aucun effet).

## enumerateParameters

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

Applique une procédure de rappel à chaque nom et valeur de paramètre d'image.

Tableau 21. Paramètres enumerateParameters

| Paramètre | Description                                                 |
|-----------|-------------------------------------------------------------|
| handle    | Descripteur de l'image.                                     |
| proc      | Procédure de rappel d'un paramètre.                         |
| data      | Données définies par l'utilisateur correspondant au rappel. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 22. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

La procédure de rappel présente le type suivant :

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

Tableau 23. Paramètres de procédure de rappel

| Paramètre | Description                                                                                  |
|-----------|----------------------------------------------------------------------------------------------|
| data      | Données définies par l'utilisateur transmises à <code>clemrtl_enumerateParameters()</code> . |
| name      | Le nom du paramètre.                                                                         |
| value     | Valeur du paramètre.                                                                         |

La procédure est appliquée une seule et unique fois pour chaque paramètre d'image, et ce dans un ordre arbitraire.

### getParameter

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

Extrait la valeur d'un paramètre d'image.

Tableau 24. Paramètres `getParameter`

| Paramètre  | Description                             |
|------------|-----------------------------------------|
| handle     | Descripteur de l'image.                 |
| name       | Le nom du paramètre.                    |
| value      | Tampon recevant la valeur de paramètre. |
| value_size | Taille du tampon de valeurs.            |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 25. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si le nom du paramètre ne correspond pas au nom d'un paramètre de l'image ou si la taille du tampon de valeurs n'est pas suffisante pour recevoir la valeur de paramètre accompagnée d'un caractère de fin nul. Les noms de paramètre utilisent le format `name.attribute` décrit ci-dessus (par exemple, `file0.name`) et les valeurs de paramètre, quant à elles, sont toujours des chaînes, même dans le cas d'une interprétation numérique.

### setParameter

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

Définit la valeur d'un paramètre d'image.

Tableau 26. Paramètres *setParameter*

| Paramètre | Description             |
|-----------|-------------------------|
| handle    | Descripteur de l'image. |
| name      | Le nom du paramètre.    |
| value     | Valeur du paramètre.    |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 27. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si le descripteur d'image n'est pas valide ou si le nom de paramètre ne correspond pas au nom d'un paramètre de l'image. Les noms de paramètre utilisent le format `name.attribute` décrit ci-dessus (par exemple, `file0.name`) et les valeurs de paramètre, quant à elles, sont toujours des chaînes, même dans le cas d'une interprétation numérique.

### getFieldCount

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

Renvoie le nombre de champs dans une source d'entrée ou une cible de sortie.

Tableau 28. Paramètres *getFieldCount*

| Paramètre   | Description                                                                                                                                                                     |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handle      | Descripteur de l'image.                                                                                                                                                         |
| key         | Le nom de l'entrée ou de la sortie à examiner, tel qu'il se trouve dans le fichier des paramètres. La clé peut également faire référence à un fichier ou à une base de données. |
| field_count | Reçoit le nombre de champs.                                                                                                                                                     |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 29. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

### getFieldTypes

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

Renvoie les types de champs d'une source d'entrée ou d'une cible de sortie.

Tableau 30. Paramètres `getFieldTypes`

| Paramètre                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>handle</code>      | Descripteur de l'image.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>key</code>         | Le nom de l'entrée ou de la sortie à examiner, tel qu'il se trouve dans le fichier des paramètres. La clé peut également faire référence à un fichier ou à une base de données.                                                                                                                                                                                                                                                                                     |
| <code>field_count</code> | Le nombre de champs à examiner. Ce nombre doit être inférieur ou égal à la longueur de l'ensemble de <code>types_champ</code> .                                                                                                                                                                                                                                                                                                                                     |
| <code>field_types</code> | Un ensemble d'une longueur d'au moins <code>comptage_champ</code> qui reçoit les types de champs. Les types des premiers champs <code>comptage_champ</code> dans l'entrée ou la sortie sont copiés dans l'ensemble. Si le <code>comptage_champ</code> est supérieur au nombre réel de champs, les éléments supplémentaires de l'ensemble ne sont pas définis. Les valeurs des types de champ se trouvent dans une liste du tableau des types de données ci-dessous. |

## Types de données

Les valeurs possibles pour `field_types` sont présentées dans le tableau ci-après.

Tableau 31. Types de données

| Type       | Interprétation                                            | Déclarateur 'C' standard |
|------------|-----------------------------------------------------------|--------------------------|
| CHAINE     | Chaîne de caractères de fin nuls UTF-8.                   | <code>const char*</code> |
| INTEGER    | Entier signé 32 bits.                                     | <code>int</code>         |
| LONG       | Entier signé 64 bits.                                     | <code>long long</code>   |
| REAL       | Virgule flottante 64 bits.                                | <code>double</code>      |
| HEURE      | Entier signé 64 bits (secondes depuis minuit).            | <code>long long</code>   |
| DATE       | Entier signé 64 bits (secondes depuis minuit 01/01/1970). | <code>long long</code>   |
| HORODATAGE | Entier signé 64 bits (secondes depuis minuit 01/01/1970). | <code>long long</code>   |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 32. Codes de statut

| Résultat                   | Description                                   |
|----------------------------|-----------------------------------------------|
| <code>CLEMRTL_OK</code>    | Réussite.                                     |
| <code>CLEMRTL_FAIL</code>  | Echec et absence de détails supplémentaires.  |
| <code>CLEMRTL_ERROR</code> | Echec et présence de détails supplémentaires. |

## `setAlternativeInput`

```
int clemrtl_setAlternativeInput(
 clemrtl_image_handle handle,
 const char* key,
```

```

size_t field_count,
int field_types,
void** (*iterator)(void* arg),
void* arg);

```

Remplace une source d'entrée de fichier par une autre source d'entrée.

Tableau 33. Paramètres *setAlternativeInput*

| Paramètre   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handle      | Descripteur de l'image.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| key         | Le nom de la source d'entrée d'origine tel qu'il est utilisé dans le fichier des paramètres. La source d'entrée doit être un fichier afin que la clé ait toujours la forme "fileN" pour l'entier $N \geq 0$ .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| field_count | Le nombre de champs dans l'entrée. La valeur doit correspondre au nombre exact de champs dans l'entrée de fichier d'origine ou l'appel échouera.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| field_types | Un ensemble de longueur d'au moins field_count. La valeur field_types[i] est le type du ième champ et doit être l'une des valeurs de la liste du tableau des types de champ ci-dessous. Ces types doivent être compatibles avec ceux de l'entrée du fichier d'origine ou l'appel échouera.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| iterator    | <p>Une fonction qui produit les autres données d'entrée. Cette fonction s'applique à ses arguments, de la façon suivante :</p> <pre>void** row = iterator(arg);</pre> <p>Cette fonction est appelée pendant l'exécution (dans un appel à <code>clemrtl_execute</code>) est appelée une fois par enregistrement d'entrée. Une valeur de renvoi NULL indique la fin de l'entrée auquel cas la fonction n'est pas rappelée et l'exécution se termine. Sinon, le résultat est un ensemble de données de longueur d'au moins field_count où row[i] donne la valeur du ième champ. Une valeur peut être NULL, où alors elle doit être un pointeur vers une donnée dont le type est déterminé par les field_types[i] correspondants. Les pointeurs doivent être valides jusqu'au prochain appel à l'itérateur ou jusqu'à la fin de l'exécution si l'exécution se termine prématurément.</p> |
| arg         | Un argument opaque transmis à l'itérateur à chaque appel.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## Types de données

Les field\_types doivent appartenir à une liste de types restreinte. Pour plus d'informations, voir la rubrique «getFieldTypes», à la page 30.

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 34. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si les types de champ ne sont pas compatibles avec l'entrée de fichier d'origine, ou si le comptage de champ ne correspond pas au nombre exact de champs dans l'entrée de fichier d'origine.



## 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);
```

Remplace une cible de sortie de fichier par une autre cible de sortie.

Tableau 35. Paramètres setAlternativeOutput

| Paramètre   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handle      | Descripteur de l'image.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| key         | Le nom de la cible de sortie d'origine tel qu'il est utilisé dans le fichier des paramètres. La cible de sortie doit être un fichier afin que la clé ait toujours la forme "fileN" pour l'entier $N \geq 0$ .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| field_count | Le nombre de champs dans la sortie. La valeur doit correspondre au nombre exact de champs dans la sortie de fichier d'origine ou l'appel échouera.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| field_types | Un ensemble de longueur d'au moins field_count. La valeur field_types[i] est le type du ième champ et doit être l'une des valeurs de la liste du tableau des types de champ ci-dessous. Ces types doivent être compatibles avec ceux de la sortie du fichier d'origine ou l'appel échouera.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| iterator    | Une fonction qui consomme la sortie d'image. Cette fonction s'applique à ses arguments et à une ligne de données, de la façon suivante :<br><br>iterator(arg, ligne);<br><br>Cette fonction est appelée pendant l'exécution (dans un appel à clemrtl_execute) et est appelée une fois par ligne de résultats produite par l'image. Une valeur ligne NULL indique la fin de la sortie après laquelle la fonction n'est plus appelée. Une application ne doit pas supposer ce dernier appel et doit vider puis fermer toute ressource externe, etc. lorsque l'exécution est terminée. Sinon, row est un ensemble de données de longueur d'au moins field_count où row[i] renvoie la valeur du ième résultat. Une valeur peut être NULL, où alors elle doit être un pointeur vers une donnée dont le type est déterminé par les field_types[i] correspondants. Cette fonction doit copier toutes les valeurs de données nécessaires parce que la mémoire pourrait ne pas être conservée après le renvoi de l'appel. |
| arg         | Un argument opaque transmis à l'itérateur à chaque appel.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

### Types de données

Les field\_types doivent appartenir à une liste de types restreinte. Pour plus d'informations, voir la rubrique «getFieldTypes», à la page 30.

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 36. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_FAIL  | Echec et absence de détails supplémentaires.  |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si les types de champ ne sont pas compatibles avec la sortie de fichier d'origine, ou si le comptage de champ ne correspond pas au nombre exact de champs dans la sortie de fichier d'origine.

### execute

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

Exécute une image.

Tableau 37. Paramètres execute

| Paramètre | Description             |
|-----------|-------------------------|
| handle    | Descripteur de l'image. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 38. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

Si l'image n'a pas été préparée, elle l'est alors à l'aide des valeurs de paramètre actuelles. L'opération échoue si le descripteur d'image n'est pas valide ou si une erreur se produit au cours de la préparation ou de l'exécution. L'appel n'est pas renvoyé tant que l'exécution n'est pas terminée.

### prepare

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

Prépare une image en vue de son exécution.

Tableau 39. Paramètres prepare

| Paramètre | Description             |
|-----------|-------------------------|
| handle    | Descripteur de l'image. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 40. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération échoue si le descripteur d'image n'est pas valide ou si une erreur se produit au cours de la préparation.

Une image doit être préparée avant d'être exécutée. La préparation d'une image gèle les valeurs de paramètre dans l'image. `clemrtl_execute()` prépare automatiquement une image sur chaque appel si l'image n'a pas déjà été préparée. L'utilisation de `clemrtl_prepare()` s'avère utile pour préparer une image lorsque celle-ci doit être exécutée plusieurs fois avec les mêmes valeurs de paramètre ; en outre, cette commande permet d'améliorer considérablement les performances. Une fois l'image préparée, les modifications suivantes apportées aux valeurs de paramètre sont ignorées ; utilisez à nouveau `clemrtl_prepare()` pour mettre à jour l'image avec les nouvelles valeurs de paramètre.

## interrupt

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

Termine l'exécution en cours sur une image.

Tableau 41. Paramètres `interrupt`

| Paramètre | Description             |
|-----------|-------------------------|
| handle    | Descripteur de l'image. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 42. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

L'opération n'a aucun effet si le descripteur d'image n'est pas valide ou si l'image n'est pas en cours d'exécution.

Il est possible d'effectuer l'appel de cette fonction en même temps qu'un autre appel d'API sur le même descripteur d'image.

## getErrorDetail

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

Extrait des informations détaillées sur la dernière erreur qui s'est produite sur une image.

Tableau 43. Paramètres `getErrorDetail`

| Paramètre | Description                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------|
| handle    | Descripteur de l'image.                                                                                                        |
| gravité   | Reçoit le code de gravité sous la forme d'un caractère unique :<br><br>I—information<br>W—warning<br>E—error<br>X—system error |
| code      | Reçoit le numéro de l'erreur.                                                                                                  |
| text      | Tampon recevant le texte du message.                                                                                           |

Tableau 43. Paramètres *getErrorDetail* (suite)

| Paramètre | Description                |
|-----------|----------------------------|
| text_size | Taille du tampon du texte. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 44. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

Les résultats ne sont pas fiables si le descripteur d'image n'est pas valide. Le texte du message est tronqué si nécessaire pour qu'il tienne dans le tampon du texte.

Si un appel d'API échoue en raison d'une erreur CLEMRTL\_ERROR et que cette erreur n'est liée à aucun descripteur d'image particulier, l'utilisation de la valeur 0 comme descripteur d'image permet d'extraire les informations de la dernière erreur non relative aux images ; toutefois, cette opération n'est pas fiable dans un environnement multi-unité d'exécution.

## setReportHandler

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

Installe une procédure de rappel de rapport pour une image.

Tableau 45. Paramètres *setReportHandler*

| Paramètre | Description                                                 |
|-----------|-------------------------------------------------------------|
| handle    | Descripteur de l'image.                                     |
| proc      | Procédure de rappel de rapport.                             |
| data      | Données définies par l'utilisateur correspondant au rappel. |

Renvoie l'un des codes de statut indiqués dans le tableau ci-après.

Tableau 46. Codes de statut

| Résultat      | Description                                   |
|---------------|-----------------------------------------------|
| CLEMRTL_OK    | Réussite.                                     |
| CLEMRTL_ERROR | Echec et présence de détails supplémentaires. |

La procédure de rappel présente le type suivant :

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

Tableau 47. Paramètres de procédure de rappel

| Paramètre | Description                                                                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------|
| data      | Données définies par l'utilisateur transmises à <code>clemrtl_setReportHandler()</code> .                        |
| severity  | Code de gravité sous la forme d'un caractère unique :<br>I—information<br>W—warning<br>E—error<br>X—system error |
| code      | Numéro du message.                                                                                               |
| text      | Texte du message.                                                                                                |

La procédure est appliquée à chaque message dès qu'il est généré. En paramétrant la procédure sur NULL, vous supprimez tout descripteur existant installé sur l'image.



---

## Remarques

Ces informations ont été développées pour les produits et services offerts dans le monde.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, programme ou service IBM n'implique pas que seul ce produit, programme ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent peut être utilisé s'il n'enfreint aucun droit de propriété intellectuelle d'IBM. Cependant l'utilisateur doit évaluer et vérifier l'utilisation d'un produit, programme ou service non IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. L'octroi de ce document n'équivaut aucunement à celui d'une licence pour ces brevets. Vous pouvez envoyer par écrit des questions concernant la licence à :

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

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Pour toute demande au sujet des licences concernant les jeux de caractères codés sur deux octets (DBCS), contactez le service Propriété intellectuelle IBM de votre pays ou adressez vos questions par écrit à :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certains états n'autorisent pas l'exclusion de garanties explicites ou implicites lors de certaines transactions, par conséquent, il est possible que cet énoncé ne vous concerne pas.

Ces informations peuvent contenir des erreurs techniques ou des erreurs typographiques. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Toute référence dans ces informations à des sites Web autres qu'IBM est fournie dans un but pratique uniquement et ne sert en aucun cas de recommandation pour ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation à votre égard, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Software Group  
ATTN: Licensing  
200 W. Madison St.  
Chicago, IL; 60606  
U.S.A.

Ces informations peuvent être disponibles, soumises à des conditions générales, et dans certains cas payantes.

Le programme sous licence décrit dans le présent document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret Contractuel IBM, des Conditions internationales d'utilisation des Logiciels IBM ou de tout autre contrat équivalent.

Toutes les données sur les performances contenues dans le présent document ont été obtenues dans un environnement contrôlé. Par conséquent, les résultats obtenus dans d'autres environnements d'exploitation peuvent varier de manière significative. Certaines mesures peuvent avoir été effectuées sur des systèmes en cours de développement et il est impossible de garantir que ces mesures seront les mêmes sur les systèmes commercialisés. De plus, certaines mesures peuvent avoir été estimées par extrapolation. Les résultats réels peuvent être différents. Les utilisateurs de ce document doivent vérifier les données applicables à leur environnement spécifique.

les informations concernant les produits autres qu'IBM ont été obtenues auprès des fabricants de ces produits, leurs annonces publiques ou d'autres sources publiques disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Aucune réclamation relative à des produits non IBM ne pourra être reçue par IBM. Les questions sur les capacités de produits autres qu'IBM doivent être adressées aux fabricants de ces produits.

Toutes les déclarations concernant la direction ou les intentions futures d'IBM peuvent être modifiées ou retirées sans avertissement préalable et représentent uniquement des buts et des objectifs.

Ces informations contiennent des exemples de données et de rapports utilisés au cours d'opérations quotidiennes standard. Pour les illustrer le mieux possible, ces exemples contiennent des noms d'individus, d'entreprises, de marques et de produits. Tous ces noms sont fictifs et toute ressemblance avec des noms et des adresses utilisés par une entreprise réelle ne serait que pure coïncidence.

Si vous consultez la version papier de ces informations, il est possible que certaines photographies et illustrations en couleurs n'apparaissent pas.

---

## Marques

IBM, le logo IBM et [ibm.com](http://ibm.com) sont des marques d'International Business Machines dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark information" à l'adresse [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques commerciales ou des marques déposées de Intel Corporation ou de ses filiales aux Etats-Unis et dans d'autres pays.

Linux est une marque déposée de Linus Torvalds aux Etats-Unis et/ou dans d'autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques commerciales de Microsoft Corporation aux Etats-Unis et/ou dans d'autres pays.

UNIX est une marque déposée de The Open Group aux Etats-Unis et dans d'autres pays.

Les marques commerciales Java et basées sur Java ainsi que les logos sont des marques commerciales ou déposées de Oracle et/ou de ses filiales.

Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.



---

# Index

## A

API  
aperçu 13  
autre entrée 31  
autre sortie 33  
codes d'erreur 23  
codes de statut 23  
comptage du champ 30  
options d'exécution 25, 26  
référence 23  
setReportHandler 36  
types de champ 30

## B

bibliothèque de programmation Runtime  
(CLEMRTL) 13

## C

closeImage  
présentation du processus d'API 23  
référence sur les fonctions API 28  
codes d'erreur 23  
codes de statut 23

## D

déploiement 7, 21  
désinstallation  
IBM SPSS Modeler Solution Publisher  
Runtime 19  
documentation 3

## E

enumerateParameters  
présentation du processus d'API 23  
référence sur les fonctions API 28  
EPM  
API 21  
execute  
référence sur les fonctions API 34  
exécution  
présentation du processus d'API 23  
exécution de flux  
utilisation d'IBM SPSS Modeler  
Solution Publisher 11, 12  
exemples  
Guide des applications 3  
présentation 5  
exemples d'application 3

## F

fenêtres  
installation d'IBM SPSS Modeler  
Solution Publisher Runtime 15

fermeture  
flux 28  
fichier .par 8  
fichier .pim 8  
fichiers journaux 26  
fonctions API  
closeImage 23, 28  
enumerateParameters 23, 28  
execute 34  
exécution 23  
getErrorDetail 23, 35  
getFieldCount 23, 30  
getFieldTypes 23, 30  
getOption 23, 25  
getParameter 23, 29  
initialisation 23, 24  
interrupt 23, 35  
openImage 23, 27  
prepare 23, 34  
setAlternativeInput 23, 31  
setAlternativeOutput 23, 33  
setLogFile 23, 26  
setOption 23, 26  
setParameter 23, 29  
setReportHandler 23  
fuseau horaire  
défini à l'aide de l'API 24

## G

getErrorDetail  
présentation du processus d'API 23  
référence sur les fonctions API 35  
getFieldCount  
présentation du processus d'API 23  
référence sur les fonctions API 30  
getFieldTypes  
présentation du processus d'API 23  
référence sur les fonctions API 30  
getOption  
présentation du processus d'API 23  
référence sur les fonctions API 25  
getParameter  
présentation du processus d'API 23  
référence sur les fonctions API 29

## I

IBM SPSS Embeddable Predictive  
Analytics  
API 21  
IBM SPSS Modeler 1  
documentation 3  
IBM SPSS Modeler Server 1  
IBM SPSS Modeler Solution Publisher 7,  
8, 21  
aperçu 7  
intégration dans des applications 13

IBM SPSS Modeler Solution Publisher  
Runtime  
configuration système requise 15, 16  
démarrage 18  
désinstallation 19  
installation 15  
répertoire temporaire 18  
initialise  
indicateurs 24  
référence sur les fonctions API 24  
initialise\_ext  
indicateurs 24  
référence sur les fonctions API 24  
initialize  
présentation du processus d'API 23  
interface de programme d'application  
aperçu 23  
interrupt  
présentation du processus d'API 23  
référence sur les fonctions API 35

## N

noeud IBM SPSS Modeler Solution  
Publisher 21  
noeud Publier 21  
noeuds de sortie 7, 21

## O

openImage  
présentation du processus d'API 23  
référence sur les fonctions API 27  
ouverture  
flux 27

## P

paramètres  
fichiers de paramètres API 27, 28  
paramètres d'exécution de flux 12  
paramètres d'image API 28, 29  
procédures de rappel d'API 28  
paramètres régionaux  
défini à l'aide de l'API 24  
prepare  
présentation du processus d'API 23  
référence sur les fonctions API 34  
publication de flux  
IBM SPSS Modeler Solution  
Publisher 7, 8

## R

répertoire temporaire  
IBM SPSS Modeler Solution Publisher  
Runtime 18  
Runtime  
démarrage 18

Runtime (*suite*)  
désinstallation 19  
installation sous UNIX 16  
installation sous Windows 16  
options 11  
répertoire temporaire 18

## S

setAlternativeInput  
présentation du processus d'API 23  
référence sur les fonctions API 31  
setAlternativeOutput  
présentation du processus d'API 23  
référence sur les fonctions API 33  
setLogFile  
présentation du processus d'API 23  
référence sur les fonctions API 26  
setOption  
présentation du processus d'API 23  
référence sur les fonctions API 26  
setParameter  
présentation du processus d'API 23  
référence sur les fonctions API 29  
setReportHandler  
présentation du processus d'API 23  
référence sur les fonctions API 36  
signalisation des erreurs  
référence sur les fonctions API 26,  
35, 36

## U

UNIX  
installation d'IBM SPSS Modeler  
Solution Publisher Runtime 16



