

IBM SPSS Modeler 14.2 Solution  
Publisher



*Remarque* : avant d'utiliser ces informations et le produit dont elles traitent, consultez les informations générales dans la rubrique Avis sur p. .

Ce document contient des informations sur la propriété de SPSS Inc., an IBM Company. Il est fourni sous accord de licence et est protégé par la loi sur le copyright. Les informations contenues dans cette publication n'incluent pas les garanties de produit et aucune des provisions de ce manuel ne devra être interprétée comme tel.

Lorsque vous envoyez des informations à IBM ou SPSS, vous octroyez à IBM et SPSS le droit non exclusif d'utiliser ou de distribuer ces informations de la façon dont IBM ou SPSS le juge approprié sans aucune obligation envers vous.

© **Copyright IBM Corporation 1994, 2011..**

---

# Préface

IBM® SPSS® Modeler est le puissant utilitaire de Data mining 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 dispose 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 de IBM Business Analytics***

Le logiciel IBM Business Analytics fournit des informations complètes, cohérentes et précises que les preneurs de décision utilisent avec confiance pour améliorer la performance du marché. Un portefeuille étendu d'outils de [business intelligence](#), d'[analyses prédictives](#), de [performance financière et de gestion de stratégie](#), et des [applications analytiques](#) offre des connaissances claires, immédiates et applicables pour améliorer l'efficacité actuelle ainsi que la capacité de prévoir les résultats futurs. Combinées avec de riches solutions industrielles, des pratiques éprouvées et des services professionnels, les organisations de toutes tailles peuvent atteindre la productivité la plus élevée, automatiser des décisions en toute tranquillité et fournir de meilleurs résultats.

Dans le cadre de ce portefeuille, le logiciel IBM SPSS Predictive Analytics aide les organisations à prévoir des événements futurs et à agir en conséquence pour mener à de meilleurs résultats. Des clients dans le domaine commercial, gouvernemental et académique à travers le monde font confiance à la technologie IBM SPSS et considèrent qu'elle représente un avantage compétitif pour attirer, retenir et ajouter des clients, tout en réduisant la fraude et en atténuant les risques. En incorporant le logiciel IBM SPSS dans leur opérations quotidiennes, les organisations deviennent des entreprises prédictives – capables de diriger et d'automatiser les décisions pour atteindre les buts qu'ils se sont fixés et obtenir des avantages compétitifs sensibles. Pour informations supplémentaires ou pour joindre un revendeur, visitez le site <http://www.ibm.com/spss>.

## ***Assistance technique***

L'assistance technique est à la disposition des clients pour la maintenance des produits. 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 joindre l'assistance technique, consultez le site Web de IBM Corp. à l'adresse <http://www.ibm.com/support>. Lorsque vous contactez l'assistance technique, n'oubliez pas de préparer vos identifiants, le nom de votre société et votre contrat d'assistance.

---

# Contenu

<b>1</b>	<b><i>A propos de IBM SPSS Modeler</i></b>	<b>1</b>
	IBM SPSS Modeler Server . . . . .	1
	IBM SPSS Modeler Options . . . . .	2
	IBM SPSS Text Analytics . . . . .	2
	Documentation de IBM SPSS Modeler . . . . .	2
	Exemples d'application . . . . .	4
	Dossier Demos . . . . .	4
<b>2</b>	<b><i>IBM SPSS Modeler Solution Publisher</i></b>	<b>6</b>
	Fonctionnement de IBM SPSS Modeler Solution Publisher . . . . .	6
	Publication de flux . . . . .	7
<b>3</b>	<b><i>Exécution de flux publiés</i></b>	<b>10</b>
	Utilisation du fichier de paramètres pour personnaliser l'exécution de flux . . . . .	11
	Intégration de IBM SPSS Modeler Solution Publisher dans les applications . . . . .	12
	Exemple d'application IBM SPSS Modeler Solution Publisher . . . . .	13
<b>4</b>	<b><i>Installation de IBM SPSS Modeler Solution Publisher Runtime</i></b>	<b>15</b>
	Installation de Runtime . . . . .	15
	Configuration requise . . . . .	15
	Installation sous Windows . . . . .	16
	Installation sous UNIX . . . . .	16
	Démarrage du programme Runtime . . . . .	19
	Changement de répertoire temporaire . . . . .	20
	Désinstallation de IBM SPSS Modeler Solution Publisher Runtime . . . . .	20
	Procédure de désinstallation sous Windows . . . . .	20
	Procédure de désinstallation sous UNIX . . . . .	20

**5 Référence de l'API de la bibliothèque d'exécution IBM SPSS  
Modeler Solution Publisher 21**

Introduction ..... 21  
Présentation du processus d'API ..... 21  
Fonctions API ..... 22

**Annexe**

**A Avis 36**

**Index 39**



# *A propos de IBM SPSS Modeler*

IBM® SPSS® Modeler est un ensemble d'outils de data mining 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, SPSS Modeler prend en charge l'intégralité du processus de Data mining, des données à l'obtention de meilleurs résultats commerciaux.

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 combinaison avec SPSS Modeler Server. Plusieurs autres options sont également disponibles, telles que décrites dans les sections suivantes. Pour plus d'informations, consultez <http://www.ibm.com/software/analytics/spss/products/modeler/>.

## ***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 ensembles de données plus volumineux. D'autres mises à jour ou produits que ceux mentionnés ici peuvent également être disponibles. Pour plus d'informations, consultez <http://www.ibm.com/software/analytics/spss/products/modeler/>.

**SPSS Modeler. La version** SPSS Modeler intègre toutes les fonctions du produit. Elle est installée et exécutée sur l'ordinateur de bureau de l'utilisateur. Pour obtenir de meilleures performances lors du traitement d'ensembles de données volumineux, vous pouvez l'exécuter en mode local, comme produit autonome, ou en mode réparti, en association avec IBM® SPSS® Modeler Server.

**SPSS Modeler Server.** SPSS Modeler Server s'exécute en continu en mode d'analyse réparti, conjointement avec une ou plusieurs installations IBM® SPSS® Modeler, obtenant ainsi de meilleures performances lors du traitement des ensembles de données volumineux. En effet, les opérations consommatrices de mémoire s'effectuent sur le serveur : inutile de télécharger les données sur l'ordinateur client. SPSS Modeler Server offre également une prise en charge de l'optimisation SQL et des capacités de modélisation au sein de la base de données, ce qui présente de nouveaux avantages en matière de performances et d'automatisation. Pour lancer les analyses, vous devez disposer d'au moins une installation SPSS Modeler.

## **IBM SPSS Modeler Options**

Vous pouvez acheter et obtenir une licence pour les fonctions et composants suivants à utiliser avec SPSS Modeler. Notez que des produits ou des mises à jour supplémentaires deviennent parfois disponibles. Pour plus d'informations, consultez <http://www.ibm.com/software/analytics/spss/products/modeler/>.

- Accès à SPSS Modeler Server, assurant une meilleure extensibilité et des performances accrues pour les ensembles de données volumineux, ainsi qu'une prise en charge de l'optimisation SQL et des capacités de modélisation au sein de la base de données.
- SPSS Modeler Solution Publisher, pour le scoring en temps réel ou automatique en dehors de l'environnement SPSS Modeler. [Pour plus d'informations, reportez-vous à la section IBM SPSS Modeler Solution Publisher dans le chapitre 2 sur p. 6.](#)
- Adaptateurs permettant le déploiement dans IBM SPSS Collaboration and Deployment Services ou dans l'application client léger IBM SPSS Modeler Advantage. [Pour plus d'informations, reportez-vous à la section Stockage et et déploiement des objets IBM SPSS Collaboration and Deployment Services Repository dans le chapitre 9 dans \*Guide de l'utilisateur de IBM SPSS Modeler 14.2.\*](#)

## **IBM SPSS Text Analytics**

IBM® SPSS® Text Analytics est un complément totalement intégré pour SPSS Modeler, qui utilise des technologies linguistiques avancées et le traitement du langage naturel pour traiter rapidement un grand nombre de données texte non structurées, extraire et organiser les concepts-clés et grouper ces concepts en 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 de Data mining de IBM® SPSS® Modeler, afin de favoriser une prise de décision précise et efficace.

- Le Text Mining offre une modélisation des concepts et des catégories, ainsi qu'un utilitaire interactif où vous pouvez exécuter une exploration avancée des liens textuels et des classes, créer vos propres catégories et affiner les modèles de ressources linguistiques.
- De nombreux formats d'importation sont pris en charge, y compris les blogs et sources Web.
- Enfin, des modèles personnalisés, des bibliothèques et des dictionnaires spécialisés dans des domaines précis, tels que la gestion de la relation client et la génomique, sont fournis.

*Remarque* : Une licence distincte est requise pour accéder à ce composant. Pour plus d'informations, consultez <http://www.ibm.com/software/analytics/spss/products/modeler/>.

## **Documentation de IBM SPSS Modeler**

Une documentation complète au format d'aide en ligne est disponible dans le menu Aide de SPSS Modeler. Vous y trouverez la documentation de SPSS Modeler, SPSS Modeler Server et de SPSS Modeler Solution Publisher, ainsi que le Guide des applications et d'autres documentations utiles.

La documentation complète de chaque produit au format PDF est disponible dans le dossier *\Documentation* de chaque DVD de produit.

- **Guide de l'utilisateur 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 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.
- **IBM SPSS Modeler Noeuds de modélisation.** Description de tous les noeuds utilisés pour créer des modèles de Data mining. IBM® SPSS® Modeler propose différentes méthodes de modélisation issues des domaines de l'apprentissage automatique, de l'intelligence artificielle et des statistiques. [Pour plus d'informations, reportez-vous à la section Description des noeuds de modélisation dans le chapitre 3 dans Noeuds de modélisation IBM SPSS Modeler 14.2.](#)
- **Guide des Algorithmes IBM SPSS Modeler.** Descriptions des fondements mathématiques des méthodes de modélisation utilisées dans SPSS Modeler.
- **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, reportez-vous à la section Exemples d'application dans Guide de l'utilisateur de IBM SPSS Modeler 14.2.](#)
- **Génération de scripts et automatisation IBM SPSS Modeler.** Informations sur l'automatisation du système via la génération de scripts, y compris les propriétés permettant de manipuler les noeuds et les flux.
- **IBM SPSS Modeler Guide de déploiement.** Informations sur l'exécution des scénarios et des flux SPSS Modeler comme étapes des tâches d'exécution sous IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **IBM SPSS Modeler CLEF Guide du développeur.** 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 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 fonctions analytiques via des algorithmes tiers.
- **IBM SPSS Modeler Server et Guide des performances.** Informations sur le mode de configuration et d'administration de IBM® SPSS® Modeler Server.
- **Guide de l'utilisateur de IBM SPSS Modeler Administration Console.** Informations concernant l'installation et l'utilisation de l'interface utilisateur de la console permettant de surveiller et de configurer SPSS Modeler Server. La console est implémentée en tant que plug-in à l'application Deployment Manager.
- **Guide IBM SPSS Modeler Solution Publisher.** SPSS Modeler Solution Publisher est un module complémentaire qui permet aux entreprises de publier des flux destinés à être utilisés en dehors de l'environnement SPSS Modeler.
- **Guide CRISP-DM IBM SPSS Modeler** Guide détaillé sur l'utilisation de la méthodologie CRISP-DM pour le Data mining avec SPSS Modeler

## **Exemples d'application**

Tandis que les outils de Data mining de SPSS Modeler peuvent vous aider à résoudre une grande variété de problèmes commerciaux et organisationnels, les exemples d'application fournissent des introductions brèves et ciblées aux méthodes et aux techniques de modélisation. Les ensembles 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, reportez-vous à la section Dossier Demos dans \*Guide de l'utilisateur de IBM SPSS Modeler 14.2\*.](#)

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

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

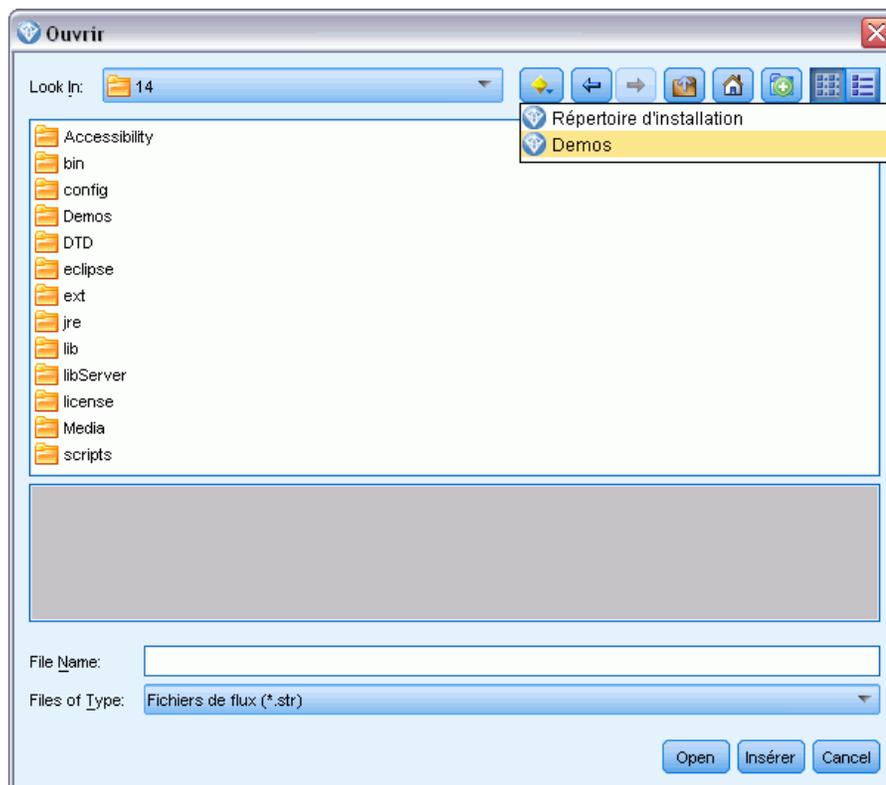
## **Dossier Demos**

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 sous IBM SPSS Modeler 14.2 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.

Figure 1-1

Sélection du dossier *Demos* dans la liste des répertoires récemment consultés



# **IBM SPSS Modeler Solution Publisher**

IBM® SPSS® Modeler Solution Publisher est un puissant outil qui permet d'intégrer les résultats de Data mining dans votre processus commercial afin de résoudre des problèmes réels. SPSS Modeler Solution Publisher permet de 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 commerciaux quotidiens et aider les décideurs de l'entreprise grâce aux connaissances issues de l'exploration des données.

Grâce à 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, reportez-vous à la section Publication de flux sur p. 7.](#) 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.

## **Licences IBM SPSS Modeler Solution Publisher**

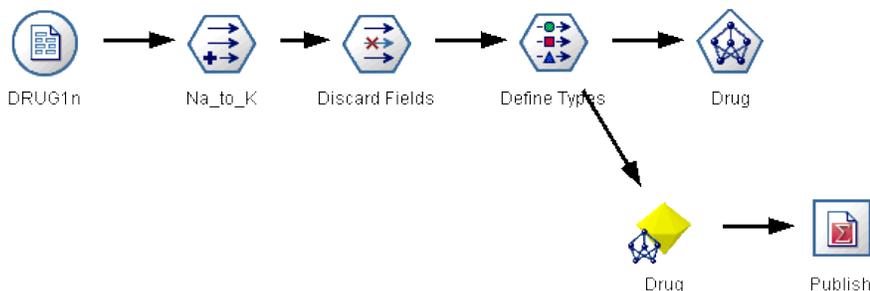
*Remarque* : SPSS Modeler Solution Publisher est distribué en tant que partie intégrante de IBM® SPSS® Collaboration and Deployment Services Scoring Service, produit pour lequel une licence distincte est requise. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

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, reportez-vous à la section Installation de Runtime dans le chapitre 4 sur p. 15.](#)

## **Fonctionnement de IBM SPSS Modeler Solution Publisher**

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

Figure 2-1  
Publication d'un flux



**Publication.** Après avoir travaillé sur le processus de Data mining, 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 commercial. 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, reportez-vous à la section Publication de flux sur p. 7.](#)

**Exécution.** 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 (*modellerrun.exe*) ou développez une application utilisant la bibliothèque SPSS Modeler Runtime pour exécuter le flux. Pour exécuter des flux à l'extérieur de 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, reportez-vous à la section Installation de Runtime dans le chapitre 4 sur p. 15.](#)

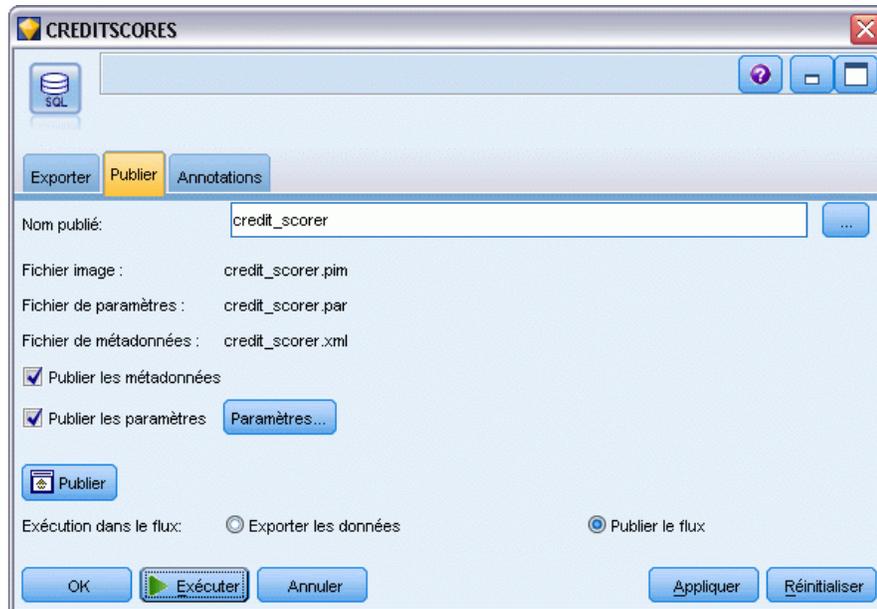
## Publication de flux

La publication des flux est effectuée directement à partir de IBM® SPSS® Modeler à l'aide des noeuds d'export standard : les noeuds SGBD, Fichier plat, Export Statistics, Export IBM® SPSS® Data Collection, Export SAS et Excel, noeuds d'exportation 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 d'export SGBD. [Pour plus d'informations, reportez-vous à la section Présentation des noeuds d'exportation dans le chapitre 7 dans IBM SPSS Modeler 14.2 Noeuds source, exécution et de sortie.](#)

### Pour publier un flux

- ▶ Ouvrez ou créez un flux, puis reliez un noeud d'export à la fin du flux.
- ▶ 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).
- ▶ 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.

Figure 2-2  
Onglet Publier



**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 .

**Publier les 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.

Figure 2-3  
Sélection des paramètres



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

**Exécution dans le flux.** Indique 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 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 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. [Pour plus d'informations, reportez-vous à la section Filtrage ou modification du nom des champs dans le chapitre 4 dans IBM SPSS Modeler 14.2 Noeuds source, exécution et de sortie.](#)

## Exécution de flux publiés

La génération d'un ensemble 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>
```

où

-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 <b>option=</b> value précisant les options d'exécution du programme Runtime. Les options en question sont décrites plus bas (facultatif).
<image-file>	est le nom du fichier image à exécuter (obligatoire).

Les options d'exécution suivantes sont disponibles :

Table 3-1  
*Options Runtime valides*

Name	Valeur par défaut	Description
max_file_size	-1	Taille maximale des fichiers enregistrés. La valeur -1 indique l'absence 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 nœud 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 nœuds 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.datasource`, `dbconn0.user`, etc., et les paramètres associés à la connexion de sortie, `dbconn1.datasource`, `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 :

Table 3-2  
Paramètres d'exécution

Paramètre	Rôle
<code>dbconnx.datasource</code>	Nom de source de données (DSN).
<code>dbconnx.user</code>	Nom d'utilisateur destiné aux bases de données à accès limité.
<code>dbconnx.password</code>	Mot de passe destiné aux bases de données à accès limité.
<code>dbconnx.epassword</code>	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 SPSS Modeler. Copiez et collez le mot de passe codé selon vos besoins.
<code>dbconnx.password_required</code>	Commutateur 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.
<code>dbconnx.catalog</code>	Catalogue associé à la connexion de base de données.
<code>dbtablex.name</code>	Nom du tableau de base. <i>Remarque</i> : L'ordre des champs d'un nouveau tableau doit être identique à celui des champs du premier flux publié.
<code>dbtablex.schema</code>	Nom du schéma.
<code>dbtablex.catalog</code>	Catalogue associé au tableau actuel.

Paramètre	Rôle
filex.name	Nom du fichier de données. <i>Remarque</i> : L'ordre des champs d'un nouveau fichier d'entrée doit être identique à celui des champs du premier flux publié.
filex.path	Emplacement du fichier (nom du répertoire).
filex.field_names_included	En ce qui concerne les fichiers texte, il s'agit du commutateur 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	Commutateur 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	Commutateur 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 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 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*.

### Conditions requises

Voici les plates-formes et les compilateurs testés avec la bibliothèque de programmation Runtime :

Table 3-3

*Plates-formes et compilateurs*

Plateforme	Compilateur(s) suggéré(s)
AIX	IBM XL C/C++ Enterprise Edition pour AIX version 8.0
HP-UX 11.0	aC++/ANSI C B3910B A.06.05
Solaris 64 bits	Sun Studio 11 5.8 correctif 121017-06
Linux 32 bits et 64 bits	gcc 3.2.3, Intel C++ Compiler 9.1.043
Windows	Visual Studio 2005

*Remarque* : Sur la plate-forme HP-UX, l'application doit être créée avec le commutateur de compilateur `aCC -AA` pour permettre d'établir des liens vers les bibliothèques CLEMRTL.

### Exemple d'application IBM SPSS Modeler Solution Publisher

L'application fournie à titre d'exemple, *dlltest.c*, est incluse dans le dossier *\clemrtl\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 `\clemtr\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.

# ***Installation de IBM SPSS Modeler Solution Publisher Runtime***

## ***Installation de Runtime***

Lorsque vous faites l'acquisition de IBM® SPSS® Modeler Solution Publisher en tant que module complémentaire pour IBM® SPSS® Modeler, vous recevez un disque d'installation distinct. Utilisez ce disque pour installer IBM® SPSS® Modeler Solution Publisher Runtime, application grâce à laquelle vous pouvez 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.

*Remarque* : La fonctionnalité qui vous permet de publier des flux dans SPSS Modeler est installée avec IBM® SPSS® Modeler et peut être activée grâce à l'achat d'un code d'autorisation spécial. Les fichiers publiés doivent être réexportés avec chaque version de SPSS Modeler. Par exemple, un fichier exporté à l'aide d'une version précédente ne fonctionnera pas avec la version 13 de 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 R2 pour systèmes 64 bits x64 ; Windows Server 2003 Standard Edition R2 pour systèmes 32 bits x86 ou 64 bits x64  
IBM AIX® 6.1 ou 7.1 pour systèmes POWER 64 bits ; HP-UX 11i V3 pour systèmes Itanium 64 bits ; Oracle Solaris™ 9.x ou 10 pour systèmes SPARC 64 bits ; Red Hat Enterprise Linux 5.x pour systèmes 32 bits x86 ; Red Hat Enterprise Linux 5.x ou 6.x pour systèmes 64 bit x64 ou systèmes IBM ; Red Hat Enterprise Linux Advanced Platform 5.x pour systèmes 32 bits x86 ; Red Hat Enterprise Linux Advanced Platform 5.x ou 6.x pour systèmes 64 bits x64 ou IBM System z ; SuSE Linux Enterprise Server 10 ou 11 pour systèmes 64 bits x64 ou IBM System z
- **Matériel. Processeur** Pentium® ou de type Pentium ou supérieur (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 ; Processeur Itanium pour HP-UX ; UltraSPARC II (ou supérieur) pour Solaris ; processeur Pentium ou de type Pentium ou supérieur pour Linux 32 bits ; gamme de processeurs x64 (AMD 64 et EM64T) ou IBM System z pour Linux 64 bits
- **Espace disque libre minimum.** 1 giga-octet (Go) est recommandé. 2 Go recommandés si le système est exécuté sur HP-UX.

- **Mémoire RAM.** 4 GO sont 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 Linux pour IBM System z
  - VMWare ESX Server 4,1
  - VMWare vSphere 4.0
- **Environnement de classe.** Les environnements de classe suivants prennent en charge 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.

- ▶ Insérez le disque d'installation IBM SPSS Modeler Solution Publisher dans le lecteur.
- ▶ Dans le menu d'installation, choisissez Installer IBM® SPSS® Modeler Solution Publisher Runtime ou Installer IBM® SPSS® Modeler Solution Publisher Runtime pour Win64, en fonction du système sur lequel vous choisissez d'exécuter l'installation.
- ▶ Dans l'assistant d'installation, cliquez sur Suivant pour commencer.
- ▶ Suivez les instructions qui s'affichent à l'écran. Pour continuer, cliquez sur Suivant.
- ▶ 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.
- ▶ 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 SPSS Modeler Server. Pour ce faire :

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

```
ulimit -d unlimited
```

```
ulimit -m unlimited
```

```
ulimit -f unlimited
```

- Redémarrez SPSS Modeler Server.

### **Solaris.**

Outre la configuration requise mentionnée ci-dessus, les correctifs suivants sont nécessaires pour effectuer l'installation sur Solaris. Vous trouverez les correctifs à télécharger sur le site suivant : <http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage>.

- ID des correctifs Solaris 2,9 : 111711-16, 111712-16 et 111722-05 (C++)
- ID des correctifs Solaris 2,10 : 119963 (C++)

Vous devez également disposer de la bibliothèque */usr/lib/iconv/646%UTF-8* sur votre système. Si ce n'est pas le cas, installez le logiciel *SUNWiniu8* ou *SUNWiniu8x*. Cette opération permet d'installer la bibliothèque, partie intégrante des modules iconv Indic pour UTF-8.

### **Linux 32 –bits.**

Les correctifs suivants sont également nécessaires pour une exécution correcte. Vous pouvez télécharger ces logiciels à partir du réseau Red Hat Network. Il se peut que vous deviez vous enregistrer pour pouvoir accéder à ce réseau (<https://rhn.redhat.com/network/software/search.pxt>). Utilisez la fonction de recherche pour rechercher et accéder aux différentes versions des logiciels.

- Le logiciel *libstdc++* (version : *libstdc++-3.2.3-X.Y.i386*, où *X.Y* correspond à la version 22.1 ou ultérieure).
- Le logiciel *libgcc* (version : *libgcc-3.2.3-X.Y.i386*, où *X.Y* correspond à la version 22.1 ou ultérieure).
- Le logiciel *compat-libstdc++* (version : *compat-libstdc++-33-3.2.3-X.Y.i386*, où *X.Y* correspond à la version 47.3 ou ultérieure).

### **Linux 64 –bits.**

Les correctifs suivants sont également nécessaires pour une exécution correcte. Vous pouvez télécharger ces logiciels à partir du réseau Red Hat Network. Il se peut que vous deviez vous enregistrer pour pouvoir accéder à ce réseau (<https://rhn.redhat.com/network/software/search.pxt>). Utilisez la fonction de recherche pour rechercher et accéder aux différentes versions des logiciels.

- Le logiciel *libstdc++* (version : *libstdc++-3.2.3-X.Y.x86\_64*, où *X.Y* correspond à la version 22.1 ou ultérieure).

- Le logiciel *libgcc* (version : *libgcc-3.2.3-X.Y.x86\_64*, où *X.Y* correspond à la version 22.1 ou ultérieure).
- Le logiciel *compat-libstdc++* (version : *compat-libstdc++-33-3.2.3-X.Y.x86\_64*, où *X.Y* correspond à la version 47.3 ou ultérieure).

### AIX.

En plus de la configuration système requise mentionnée ci-dessus, l'installation sous AIX requiert aussi les éléments suivants :

- XL C++ runtime doit être au moins au niveau 8.0.0.9 (lspp -l xlC.rte). Les ensembles de fichiers qui contiennent les bibliothèques sont appelées IBM C++ Runtime Environment Components for AIX et peuvent être téléchargés à l'adresse <http://www.ibm.com/software/awdtools/xlcpp/support/>.
- XL FORTRAN runtime doit être au moins au niveau 10.1.0.4 (lspp -l xlfrte). Les ensembles de fichiers qui contiennent les bibliothèques sont appelées XL Fortran Enterprise Edition V10.1 for AIX Runtime Environment et peuvent être téléchargés à l'adresse <http://www.ibm.com/software/awdtools/fortran/xlfortran/support/>.

### HP-UX.

Outre la configuration requise mentionnée ci-dessus, l'installation sur HP-UX requiert également l'installation des bibliothèques C++ Runtime. Vous pouvez les télécharger sous la forme de correctif PHSS\_26946, comme l'indique le tableau suivant. Pour accéder aux correctifs, visitez le site suivant : <http://www4.itrc.hp.com/service/patch/mainPage.do>. L'enregistrement est obligatoire mais gratuit.

## 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.

- ▶ 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.
- ▶ Remplacez les répertoires par le point de montage du DVD.
- ▶ Remplacez les répertoires par le répertoire *modelrun*.
- ▶ Modifiez pour le répertoire de la plateforme approprié.
- ▶ Exécutez le script d'installation *.bin*. Par exemple :
 

```
./modelersolutionpublisherhpia64.bin -i console
```
- ▶ Les détails de l'introduction apparaissent. Appuyez sur Entrée pour continuer.
- ▶ Les informations de licence apparaissent. Lisez la licence, saisissez 1 pour l'accepter puis appuyez sur Entrée pour continuer.

- ▶ Une liste des langues disponibles apparaît. Saisissez le numéro de la langue à installer et appuyez sur Entrée pour continuer.
- ▶ 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.
- ▶ Une invite vous demande de confirmer l'emplacement de l'installation. Lorsque celui-ci est correct, tapez y et appuyez sur Entrée.
- ▶ Un récapitulatif de préinstallation apparaît pour confirmer vos entrées. Appuyez sur Entrée pour continuer.
- ▶ Un message apparaît pour vous informer que la routine d'installation est prête à être exécutée. Appuyez sur Entrée pour continuer.
- ▶ 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.

## 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, reportez-vous à la section Publication de flux dans le chapitre 2 sur p. 7.](#) Lancez le moteur d'exécution 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 <fichier de paramètres> -o <options> <fichier image>
```

où

<parameter-file>	correspond au nom du fichier de paramètres publié (facultatif).
<options>	est une liste séparée par des virgules de paires <code>option=value</code> précisant les options d'exécution du programme Runtime.
<image file>	est le nom du fichier image publié ( <code>*.pim</code> ) 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`.

[Pour plus d'informations, reportez-vous à la section Exécution de flux publiés dans le chapitre 3 sur p. 10.](#)

## **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 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 :

- ▶ 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**

- ▶ Dans le Panneau de configuration de Windows, sélectionnez Ajout/Suppression de programmes.
- ▶ Dans la liste, sélectionnez IBM® SPSS® Modeler Solution Publisher <nn>.
- ▶ Cliquez sur Supprimer.
- ▶ 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.

# 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 d'état. Les valeurs de l'indicateur d'état sont les suivantes :

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 :

- ▶ Initialisez la bibliothèque à l'aide de `clemrtl_initialize_ext()`.
- ▶ Vérifiez et modifiez les options d'exécution en utilisant `clemrtl_getOption()` et `clemrtl_setOption()`.
- ▶ Ouvrez une image à l'aide de `clemrtl_openImage()` et recevez un descripteur d'image.
- ▶ Vérifiez et modifiez les paramètres d'image en utilisant `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` et `clemrtl_setParameter()`.
- ▶ 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()`.
- ▶ Fermez l'image à l'aide de `clemrtl_closeImage()`.

- ▶ Pour annuler une exécution en cours, utilisez `clemrtl_interrupt()`.
- ▶ 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 multi-thread 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 threads distincts 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(commutateurs, 0, 0);
```

Les nouveaux programmes doivent appeler *initialise\_ext*.

### ***initialise\_ext***

```
int clemrtl_initialize_ext(  
    commutateurs non signés,  
    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.*

Paramètre	Description
commutateurs	Contrôle certains aspects du processus d'initialisation. La valeur est construite en tant qu'opérateur OR sur les bits des commutateurs 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 : <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 suivantes sont reconnues :

Nom	Valeur
installation_directory	Le chemin d'accès entier au dossier d'installation IBM® SPSS® Modeler Solution Publisher Le dossier d'installation qui contient les dossiers bin etconfig.

Renvoie l'un des codes d'erreur suivants :

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().
- Définissez le paramètre régional LC\_NUMERIC sur "C".
- Définissez le nouveau gestionnaire global de manière à ce qu'il génère une exception quand la mémoire est épuisée.

Ce comportement peut être modifié en définissant des combinaisons des commutateurs suivants :

Repère	Description
CLEMRTL_INIT_NO_TZ	Ne définissez pas le fuseau horaire.
CLEMRTL_INIT_NO_LOCALE	Ne modifiez pas les paramètres régionaux.
CLEMRTL_INIT_NO_NEW_HANDLER	Ne paramétrez pas le nouveau gestionnaire.
CLEMRTL_INIT_LOCAL_NEW_HANDLER	Localisez le nouveau gestionnaire au sein de chaque appel d'API.

La bibliothèque exige que le paramètre régional LC\_NUMERIC soit paramétré sur “C” pour pouvoir réaliser une conversion correcte entre les nombres et les chaînes (le format de nombre IBM® SPSS® Modeler utilise les règles de 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-thread.

### **getOption**

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

Extrait la valeur d’une option d’exécution.

Paramètre	Description
name	Nom de l’option.
Valeur	Tampon qui reçoit la valeur de l’option.
value_size	Taille du tampon de valeurs.

Renvoie l’un des codes d’état suivants :

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 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. [Pour plus d’informations, reportez-vous à la section Exécution de flux publiés dans le chapitre 3 sur p. 10.](#)

### **setOption**

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

Définit la valeur d'une option d'exécution.

Paramètre	Description
name	Nom de l'option.
Valeur	Valeur de l'option.

Renvoie l'un des codes d'état suivants :

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 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.

Paramètre	Description
Mode	Le mode de consignation doit présenter l'une des valeurs suivantes : <b>CLEMRTL_NULL_LOG</b> — supprime les messages. <b>CLEMRTL_STDERROR_LOG</b> —envoie les messages vers la sortie d'erreur standard. <b>CLEMRTL_FILE_LOG</b> — envoie les messages vers le fichier désigné par le paramètre <code>log_file</code> .
log_file	Chaîne contenant le nom du fichier.

Renvoie l'un des codes d'état suivants :

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.

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 d'état suivants :

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.

### ***closelImage***

```
int clemrtl_closelImage(
    clemrtl_image_handle handle);
```

Ferme un descripteur d'image et libère ses ressources.

Paramètre	Description
handle	Descripteur de l'image.

Renvoie l'un des codes d'état suivants :

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_closelImage()` (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.

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

Renvoie l'un des codes d'état suivants :

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);
```

Paramètre	Description
Données	Données définies par l'utilisateur transmises à <code>clemrtl_enumerateParameters()</code> .
name	Nom du paramètre.
Valeur	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.

Paramètre	Description
handle	Descripteur de l'image.
name	Nom du paramètre.
Valeur	Tampon recevant la valeur de paramètre.
value_size	Taille du tampon de valeurs.

Renvoie l'un des codes d'état suivants :

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 paramètres utilisent le format `name.attribute` décrit ci-dessus —par exemple, `file0.name`— ; 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.

Paramètre	Description
handle	Descripteur de l'image.
name	Nom du paramètre.
Valeur	Valeur du paramètre.

Renvoie l'un des codes d'état suivants :

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 paramètres utilisent le format `name.attribute` décrit ci-dessus —par exemple, `file0.name`— ; 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.

Paramètre	Description
handle	Descripteur de l'image.
clé	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 d'état suivants :

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.

Paramètre	Description
handle	Descripteur de l'image.
clé	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	Le nombre de champs à examiner. Ce nombre doit être inférieur ou égal à la longueur de l'ensemble de types_champ.
field_types	Un ensemble d'une longueur d'au moins comptage_champ qui reçoit les types de champs. Les types des premiers champs comptage_champ dans l'entrée ou la sortie sont copiés dans l'ensemble. Si le comptage_champ 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 field\_types doivent être l'un des types suivants :

Entrez	Interprétation	Déclarateur 'C' standard
STRING	Chaîne de caractères de fin nuls UTF-8.	const char*
INTEGER	Entier signé 32 bits.	int
LONG	Entier signé 64 bits.	long long

Entrez	Interprétation	Déclarateur 'C' standard
REAL	Virgule flottante 64 bits.	double
TIME	Entier signé 64 bits (secondes depuis minuit).	long long
DATE	Entier signé 64 bits (secondes depuis minuit 01/01/1970).	long long
TIMESTAMP	Entier signé 64 bits (secondes depuis minuit 01/01/1970).	long long

Renvoie l'un des codes d'état suivants :

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

Paramètre	Description
handle	Descripteur de l'image.
clé	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 " <i>fileN</i> " 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.

Paramètre	Description
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 appelée et l'exécution se termine. Sinon, le résultat est un ensemble de données de longueur d'au moins <code>field_count</code> où <code>row[i]</code> 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 <code>field_types[i]</code> 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, reportez-vous à la section `getFieldTypes` sur p. 29.](#)

Renvoie l'un des codes d'état suivants :

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.

Paramètre	Description
handle	Descripteur de l'image.
clé	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	<p>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 :</p> <p>iterator(arg, ligne);</p> <p>Cette fonction est appelée pendant l'exécution (dans un appel à clemrtl_exécute) 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.</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, reportez-vous à la section getFieldTypes sur p. 29.](#)

Renvoie l'un des codes d'état suivants :

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.

Paramètre	Description
handle	Descripteur de l'image.

Renvoie l'un des codes d'état suivants :

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.

Paramètre	Description
handle	Descripteur de l'image.

Renvoie l'un des codes d'état suivants :

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 de 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.

Paramètre	Description
handle	Descripteur de l'image.

Renvoie l'un des codes d'état suivants :

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.

Paramètre	Description
handle	Descripteur de l'image.
severity	Reçoit le code de gravité sous la forme d'un caractère unique : I—informations W—avertissement E—erreur X—erreur système
code	Reçoit le numéro de l'erreur.
texte	Tampon recevant le texte du message.
text_size	Taille du tampon du texte.

Renvoie l'un des codes d'état suivants :

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-thread.

### ***setReportHandler***

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

Installe une procédure de rappel de rapport pour une image.

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

Renvoie l'un des codes d'état suivants :

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);
```

Paramètre	Description
Données	Données définies par l'utilisateur transmises à clemrtl_setReportHandler().
severity	Code de gravité sous la forme d'un caractère unique : I—informations W—avertissement E—erreur X—erreur système
code	Numéro du message.
texte	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.

# Avis

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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

**Le paragraphe suivant ne s'applique pas au Royaume Uni ni à aucun pays dont la réglementation locale est incompatible avec les dispositions suivantes :** SPSS INC., AN IBM COMPANY, FOURNIT CETTE PUBLICATION « EN L'ÉTAT » SANS GARANTIE D'AUCUNE SORTE, EXPLICITE OU IMPLICITE, Y COMPRIS, MAIS SANS SE LIMITER À, LES GARANTIES TACITES DE NON CONTREFAÇON, DE CONFORMITÉ OU D'APTITUDE À UN USAGE PARTICULIER. Certains états n'autorisent pas les clauses de non garantie explicites ou implicites lors de certaines transactions, par conséquent, il est possible que cette disposition ne vous concerne pas.

Ces informations peuvent contenir des erreurs techniques ou typographiques. Ces informations sont modifiées périodiquement ; ces modifications seront incluses dans les nouvelles versions de cette publication. SPSS Inc. a le droit d'apporter des améliorations et/ou des modifications au(x) produit(s) et/ou au(x) programme(s) décrits dans cette publication à tout moment et sans avertissement préalable.

Toute référence, contenue dans ces informations, à des sites Internet autres que ceux de SPSS et IBM est fournie pour des raisons de commodité uniquement et ne peut, en aucune sorte, être considérée comme de la publicité pour ces sites Internet. Le matériel de ces sites Internet ne fait pas partie du matériel de ce produit SPSS Inc. et l'utilisation de ces sites Internet se fait à vos risques et périls.

Lorsque vous envoyez des informations à IBM ou SPSS, vous octroyez à IBM et SPSS le droit non exclusif d'utiliser ou de distribuer ces informations de la façon dont IBM ou SPSS le juge approprié sans aucune obligation envers vous.

Les informations sur des produits autres que SPSS ont été obtenues auprès des fournisseurs de ces produits, à partir de leurs annonces publiées ou d'autres sources publiques disponibles. SPSS n'a pas effectué de test sur ces produits et ne peut confirmer la validité de leurs performances, de leur compatibilité ou de toute autre prétention associée aux produits autres que SPSS. Les questions concernant les performances des produits autres que SPSS devront être adressées aux fournisseurs de ces produits.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Ces informations contiennent des exemples de données et de rapports utilisées dans les opérations quotidiennes de l'entreprise. Pour les illustrer le plus précisément possible, ces exemples contiennent les noms de personnes, d'entreprises, de marques et de produits. Tous ces noms sont fictifs et toute ressemblance avec des noms et adresses utilisés par une entreprise réelle ne serait que pure coïncidence.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

### **Marques commerciales**

IBM, le logo IBM et [ibm.com](http://www.ibm.com) sont des marques commerciales de IBM Corporation, déposées dans de nombreuses juridictions à travers le monde. Une liste mise à jour des marques commerciales IBM est disponible sur Internet à l'adresse <http://www.ibm.com/legal/copytrade.shtml>.

SPSS est une marque commerciale de SPSS Inc., an IBM Company, déposées dans de nombreuses juridictions à travers le monde.

Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques déposées ou des marques commerciales de Adobe Systems Incorporated aux États-Unis et/ou dans d'autres pays.

IT Infrastructure Library est une marque déposée du Central Computer and Telecommunications Agency qui fait désormais partie du Office of Government Commerce.

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 déposées de Intel Corporation ou de ses filiales aux États-Unis ou dans d'autres pays.

Linux est une marque déposée de Linus Torvalds aux États-Unis, dans d'autres pays, ou les deux.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux États-Unis, dans d'autres pays, ou les deux.

ITIL est une marque déposée et une marque communautaire déposée du Office of Government Commerce et a été enregistrée auprès du Bureau américain des brevets et des marques de commerce.

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

Cell Broadband Engine est une marque de Sony Computer Entertainment, Inc. aux États-Unis, dans d'autres pays ou les deux et est par conséquent sous contrat de licence.

Java et les marques et logo Java sont des marques de Sun Microsystems, Inc. aux États-Unis, dans d'autres pays, ou les deux.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

D'autres noms de produits ou de services peuvent être des marques de IBM, SPSS ou d'autres entreprises.

- API
  - Aperçu, 12, 21
  - autre entrée, 30
  - autre sortie, 31
  - codes d'erreur, 21
  - codes d'état, 21
  - comptage du champ, 28
  - options d'exécution, 24
  - référence, 21
  - setReportHandler, 35
  - types de champ, 29
- avis juridiques, 36
- Bibliothèque de programmation Runtime (CLEMRTL), 12
- closeImage
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 26
- codes d'erreur, 21
- codes d'état, 21
- déploiement, 6
- désinstallation
  - IBM SPSS Modeler Solution Publisher Runtime, 20
- documentation, 2
- enumerateParameters
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 26
- execute
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 33
- exécution de flux
  - utilisation IBM SPSS Modeler Solution Publisher, 10–11
- exemples
  - Aperçu, 4
- Exemples
  - Guide des applications, 2
- exemples d'application, 2
- fermeture
  - flux, 26
- fichier *.par*, 7
- fichier *.pim*, 7
- fichiers journaux, 25
- fonctions API
  - closeImage, 21, 26
  - enumerateParameters, 21, 26
  - execute, 21, 33
  - getErrorDetail, 21, 34
  - getFieldCount, 21, 28
  - getFieldTypes, 21, 29
  - getOption, 21, 24
  - getParameter, 21, 27
  - initialisation, 21–22
  - interrupt, 21, 33
  - openImage, 21, 25
  - prepare, 21, 33
  - setAlternativeInput, 21, 30
  - setAlternativeOutput, 21, 31
  - setLogFile, 21, 25
  - setOption, 21, 24
  - setParameter, 21, 28
  - setReportHandler, 21
- fuseau horaire
  - défini à l'aide de l'API, 22
- getErrorDetail
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 34
- getFieldCount
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 28
- getFieldTypes
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 29
- getOption
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 24
- getParameter
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 27
- IBM SPSS Modeler, 1
  - documentation, 2
- IBM SPSS Modeler Solution Publisher, 6–7
  - Aperçu, 6
  - intégration dans des applications, 12
- IBM SPSS Modeler Solution Publisher Runtime
  - configuration système requise, 15–16
  - Démarrage, 19
  - désinstallation, 20
  - installation, 15
  - répertoire temporaire, 20
- IBM SPSS Text Analytics, 2
- initialise
  - commutateurs, 22
  - référence sur les fonctions API, 22
- initialise\_ext
  - commutateurs, 22
  - référence sur les fonctions API, 22
- initialize
  - présentation du processus d'API, 21
- interrupt
  - présentation du processus d'API, 21
  - référence sur les fonctions API, 33
- marques commerciales, 37

noeuds de sortie, 6

openImage

présentation du processus d'API, 21

référence sur les fonctions API, 25

Ouverture

flux, 25

Paramètres

fichiers de paramètres API, 25–26

paramètres d'exécution de flux, 11

paramètres d'image API, 26–28

procédures de rappel d'API, 26

paramètres régionaux

défini à l'aide de l'API, 22

prepare

présentation du processus d'API, 21

référence sur les fonctions API, 33

publication de flux

IBM SPSS Modeler Solution Publisher, 6–7

Répertoire temporaire

IBM SPSS Modeler Solution Publisher Runtime, 20

Runtime

Démarrage, 19

désinstallation, 20

installation sous UNIX, 16

installation sous Windows, 16

Options, 10

répertoire temporaire, 20

setAlternativeInput

présentation du processus d'API, 21

référence sur les fonctions API, 30

setAlternativeOutput

présentation du processus d'API, 21

référence sur les fonctions API, 31

setLogFile

présentation du processus d'API, 21

référence sur les fonctions API, 25

setOption

présentation du processus d'API, 21

référence sur les fonctions API, 24

setParameter

présentation du processus d'API, 21

référence sur les fonctions API, 28

setReportHandler

présentation du processus d'API, 21

référence sur les fonctions API, 35

signalisation des erreurs

référence sur les fonctions API, 25, 34–35

SPSS Modeler Server, 1

UNIX

Installation de IBM SPSS Modeler Solution Publisher

Runtime, 16

Windows

Installation de IBM SPSS Modeler Solution Publisher

Runtime, 15