

IBM SPSS Modeler 14.2 Solution
Publisher



Nota: Antes de utilizar esta información y el producto, lea la información general en Avisos el p. .

Este documento contiene información propiedad de SPSS Inc, an IBM Company. Se proporciona con un contrato de licencia y está protegido por leyes de derechos de autor. La información que contiene esta publicación no incluye garantías del producto y cualquier declaración de este manual no se debe considerar como tal.

Al enviar información a IBM o SPSS, el usuario concede a IBM y a SPSS el derecho no exclusivo de utilizar o distribuir la información de la forma que estime adecuada sin incurrir en obligaciones con el usuario.

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

Prefacio

IBM® SPSS® Modeler es el conjunto de programas de minería de datos de IBM Corp. orientado a las empresas. SPSS Modeler ayuda a las organizaciones a mejorar la relación con sus clientes y los ciudadanos a través de la comprensión profunda de los datos. Las organizaciones utilizan la comprensión que les ofrece SPSS Modeler para retener a los clientes más rentables, identificar las oportunidades de venta cruzada, atraer a nuevos clientes, detectar el fraude, reducir el riesgo y mejorar la prestación de servicios del gobierno.

La interfaz visual de SPSS Modeler invita a la pericia empresarial específica de los usuarios, lo que deriva en modelos predictivos más eficaces y la reducción del tiempo necesario para encontrar soluciones. SPSS Modeler ofrece muchas técnicas de modelado tales como pronósticos, clasificaciones, segmentación y algoritmos de detección de asociaciones. Una vez que se crean los modelos, IBM® SPSS® Modeler Solution Publisher permite su distribución en toda la empresa a los encargados de tomar las decisiones o a una base de datos.

Acerca de IBM Business Analytics

El software IBM Business Analytics ofrece información completa, coherente y precisa en la que los órganos de toma de decisiones confían para mejorar el rendimiento comercial. Un conjunto integral de [inteligencia empresarial](#), [análisis predictivo](#), [rendimiento comercial y gestión de estrategias](#), así como de [aplicaciones de análisis](#) le ofrece una información clara, inmediata e interactiva del rendimiento actual y la capacidad para predecir resultados futuros. En combinación con extensas soluciones sectoriales, prácticas probadas y servicios profesionales, las organizaciones de cualquier tamaño pueden conseguir el máximo de productividad y alcanzar mejores resultados.

Como parte de esta familia, el software de análisis predictivo de IBM SPSS ayuda a las organizaciones a predecir eventos futuros y actuar proactivamente según esa información para lograr mejores resultados comerciales. Los clientes comerciales, gubernamentales y académicos de todo el mundo confían en la tecnología de IBM SPSS como ventaja ante la competencia para atraer, retener y hacer crecer los clientes, reduciendo al mismo tiempo el fraude y reduciendo el riesgo. Al incorporar el software de IBM SPSS en sus operaciones diarias, las organizaciones se convierten en empresas predictivas, capaces de dirigir y automatizar decisiones para alcanzar los objetivos comerciales y lograr una ventaja considerable sobre la competencia. Para obtener más información o contactar con un representante, visite <http://www.ibm.com/spss>.

Asistencia técnica

La asistencia técnica está disponible para el mantenimiento de los clientes. Los clientes podrán ponerse en contacto con el servicio de asistencia técnica si desean recibir ayuda sobre la utilización de los productos de IBM Corp. o sobre la instalación en los entornos de hardware admitidos. Para ponerse en contacto con el servicio de asistencia, visite el IBM Corp. sitio Web en <http://www.ibm.com/support>. Prepárese para identificarse, identificar a su organización y su acuerdo de asistencia al solicitar asistencia.

Contenido

1	<i>Acerca de IBM SPSS Modeler</i>	1
	IBM SPSS Modeler Server	1
	Opciones de IBM SPSS Modeler	2
	IBM SPSS Text Analytics	2
	Documentación de IBM SPSS Modeler.	2
	Ejemplos de aplicaciones	4
	Carpeta Demos.	5
2	<i>IBM SPSS Modeler Solution Publisher</i>	6
	Modo de funcionamiento de IBM SPSS Modeler Solution Publisher	6
	Publicación de rutas.	7
3	<i>Ejecución de rutas publicadas</i>	10
	Uso del archivo de parámetros para personalizar la ejecución de una ruta	11
	Incrustación de IBM SPSS Modeler Solution Publisher en las aplicaciones	12
	Aplicación de muestra de IBM SPSS Modeler Solution Publisher	13
4	<i>Instalación de IBM SPSS Modeler Solution Publisher Runtime</i>	15
	Instalación de Runtime	15
	Requisitos del sistema	15
	Instalación en Windows	16
	Instalación en UNIX.	16
	Inicio de Runtime	19
	Modificación del directorio temporal	19
	Desinstalación de IBM SPSS Modeler Solution Publisher Runtime	20
	Procedimiento de desinstalación para Windows	20
	Procedimiento de desinstalación para UNIX	20

**5 Referencia API de la biblioteca de IBM SPSS Modeler Solution
Publisher Runtime 21**

Introducción	21
Conceptos básicos del proceso de API	21
Funciones API	22

Apéndice

A Avisos	36
-----------------	-----------

Índice	39
---------------	-----------

Acerca de IBM SPSS Modeler

IBM® SPSS® Modeler es un conjunto de herramientas de minería de datos que permite desarrollar rápidamente modelos predictivos mediante técnicas empresariales y utilizarlos en operaciones empresariales para mejorar la toma de decisiones. Con un diseño que sigue el modelo CRISP-DM, estándar del sector, SPSS Modeler admite el proceso completo de minería de datos, desde los propios datos hasta obtener los mejores resultados empresariales.

SPSS Modeler ofrece una gran variedad de métodos de modelado procedentes del aprendizaje automático, la inteligencia artificial y el estadístico. Los métodos disponibles en la paleta de modelado permiten derivar nueva información procedente de los datos y desarrollar modelos predictivos. Cada método tiene ciertos puntos fuertes y es más adecuado para determinados tipos de problemas.

SPSS Modeler puede adquirirse como producto independiente o utilizarse en conjunto con SPSS Modeler Server. También hay disponible cierto número de opciones adicionales que se resumen en las siguientes secciones. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/modeler/>.

IBM SPSS Modeler Server

SPSS Modeler utiliza una arquitectura de cliente/servidor para distribuir peticiones de cliente para operaciones que requieren un uso intensivo de los recursos a un software de servidor de gran potencia, lo que proporciona un rendimiento más rápido con conjuntos de datos de mayor volumen. También puede haber disponibles productos o actualizaciones adicionales que no se incluyan en esta lista. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/modeler/>.

SPSS Modeler. SPSS Modeler es una versión completamente funcional del producto que se instala y ejecuta en el ordenador de escritorio del usuario. Esta versión se puede ejecutar en modo local como un producto independiente o en modo distribuido junto con IBM® SPSS® Modeler Server para mejorar el rendimiento a la hora de trabajar con grandes conjuntos de datos.

SPSS Modeler Server. SPSS Modeler Server se ejecuta ininterrumpidamente en modo de análisis distribuido junto con una o varias instalaciones de IBM® SPSS® Modeler, lo que ofrece un mayor rendimiento cuando se trabaja con grandes conjuntos de datos, ya que las operaciones que requieren un uso intensivo de la memoria se pueden realizar en el servidor sin tener que descargar datos en el equipo cliente. SPSS Modeler Server también ofrece compatibilidad con las capacidades de optimización de SQL y modelado en la base de datos, lo que ofrece ventajas adicionales de rendimiento y automatización. Para ejecutar un análisis debe haber al menos una instalación de SPSS Modeler.

Opciones de IBM SPSS Modeler

Es posible adquirir una licencia de uso de los siguientes componentes y características que pueden utilizarse con SPSS Modeler. Recuerde que también puede haber disponibles productos o actualizaciones adicionales. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/modeler/>.

- Acceso a SPSS Modeler Server, que ofrece una mayor escalabilidad y rendimiento en conjuntos de datos grandes, así como compatibilidad con las capacidades de optimización de SQL y modelado en la base de datos.
- SPSS Modeler Solution Publisher, permite la puntuación automática o en tiempo real fuera del entorno de SPSS Modeler. [Si desea obtener más información, consulte el tema IBM SPSS Modeler Solution Publisher en el capítulo 2 el p. 6.](#)
- Adaptadores para permitir la distribución en IBM SPSS Collaboration and Deployment Services o la aplicación IBM SPSS Modeler Advantage de baja intensidad. [Si desea obtener más información, consulte el tema Almacenamiento y recuperación de objetos de IBM SPSS Collaboration and Deployment Services Repository en el capítulo 9 en *Manual de usuario de IBM SPSS Modeler 14.2*.](#)

IBM SPSS Text Analytics

IBM® SPSS® Text Analytics es un complemento totalmente integrado en SPSS Modeler que utiliza tecnologías de lingüística avanzada y NLP para procesar con rapidez una gran variedad de datos de texto sin estructurar, extraer y organizar los conceptos clave y agruparlos en categorías. Las categorías y conceptos extraídos se pueden combinar con los datos estructurados existentes, como pueden ser datos demográficos, y se pueden aplicar para modelar utilizando el conjunto completo de herramientas de minería de datos de IBM® SPSS® Modeler para tomar decisiones mejores y más certeras.

- El nodo Text Mining ofrece modelado de conceptos y categorías así como un programa interactivo donde se puede realizar una exploración avanzada de conglomerados y vínculos de texto, crear sus propias categorías y refinar las plantillas de recursos lingüísticos.
- Hay diversos formatos de importación compatibles, incluyendo blogs y otros orígenes basados en Web.
- También se incluyen plantillas, bibliotecas y diccionarios personalizados para dominios específicos, como puede ser la terminología CRM y genómica.

Nota: Es necesario disponer de una licencia independiente para acceder a este componente. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/modeler/>.

Documentación de IBM SPSS Modeler

Tiene a su disposición una completa documentación en formato de ayuda en línea desde el menú Ayuda de SPSS Modeler. Se incluye documentación para SPSS Modeler, SPSS Modeler Server y SPSS Modeler Solution Publisher, así como el Manual de aplicaciones y otros materiales de apoyo.

La documentación completa de cada producto en formato PDF está disponible en la carpeta *Documentation* en cada DVD del producto.

- **Manual del usuario de IBM SPSS Modeler.** Introducción general sobre cómo usar SPSS Modeler, incluyendo cómo crear rutas de datos, tratar valores perdidos, crear expresiones CLEM, trabajar con proyectos e informes y empaquetar rutas para su distribución en IBM SPSS Collaboration and Deployment Services, Predictive Applications o IBM SPSS Modeler Advantage.
- **Nodos Origen, Proceso y Resultado de IBM SPSS Modeler.** Descripciones de todos los nodos utilizados para leer, procesar y dar salida a datos en diferentes formatos. En la práctica, esto implica todos los nodos que no sean nodos de modelado.
- **Nodos de modelado de IBM SPSS Modeler.** Descripciones de todos los nodos utilizados para crear modelos de minería de datos. IBM® SPSS® Modeler ofrece una variedad de métodos de modelado tomados del aprendizaje de las máquinas, la inteligencia artificial y la estadística. [Si desea obtener más información, consulte el tema Conceptos básicos sobre nodos de modelado en el capítulo 3 en *Nodos de modelado de IBM SPSS Modeler 14.2*.](#)
- **Manual de algoritmos de IBM SPSS Modeler.** Descripciones de los fundamentos matemáticos de los métodos de modelado que se utilizan en SPSS Modeler.
- **Manual de aplicaciones de IBM SPSS Modeler.** Los ejemplos de esta guía ofrecen introducciones breves y concisas a métodos y técnicas de modelado específicos. También tiene a su disposición una versión en línea de este manual en el menú Ayuda. [Si desea obtener más información, consulte el tema Ejemplos de aplicaciones en *Manual de usuario de IBM SPSS Modeler 14.2*.](#)
- **Procesos y automatización de IBM SPSS Modeler.** Información sobre la automatización del sistema a través de procesos, incluidas las propiedades que se pueden utilizar para manipular nodos y rutas.
- **IBM SPSS Modeler Manual de distribución.** Información sobre la ejecución de rutas y escenarios de SPSS Modeler como pasos en trabajos de procesamiento en IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **Guía del desarrollador de IBM SPSS Modeler CLEF.** CLEF permite integrar programas de otros fabricantes, como rutinas de procesamiento de datos o algoritmos de modelado como nodos en SPSS Modeler.
- **Manual de minería interna de bases de datos de IBM SPSS Modeler.** Este manual incluye información sobre cómo utilizar la potencia de su base de datos, tanto para mejorar su rendimiento como para ampliar su oferta de capacidades analíticas a través de algoritmos de terceros.
- **Guía de IBM SPSS Modeler Server y su rendimiento.** Información sobre la configuración y administración de IBM® SPSS® Modeler Server.
- **Manual del usuario de IBM SPSS Modeler Administration Console.** Información sobre cómo instalar y utilizar la interfaz de usuario de la consola para supervisar y configurar SPSS Modeler Server. La consola se implementa como complemento de la aplicación Deployment Manager.

- **Manual de IBM SPSS Modeler Solution Publisher.** SPSS Modeler Solution Publisher es un componente complementario que permite a las organizaciones publicar rutas para su uso fuera del entorno estándar de SPSS Modeler.
- **Manual de CRISP-DM de IBM SPSS Modeler.** Manual que explica paso a paso cómo utilizar la metodología de CRISP-DM en la minería de datos con SPSS Modeler.

Ejemplos de aplicaciones

Mientras que las herramientas de minería de datos de SPSS Modeler pueden ayudar a resolver una amplia variedad de problemas organizativos y empresariales, los ejemplos de la aplicación ofrecen introducciones breves y adaptadas de técnicas y métodos de modelado específicos. Los conjuntos de datos utilizados aquí son mucho más pequeños que los enormes almacenes de datos gestionados por algunos analizadores de datos, pero los conceptos y métodos implicados deberían ser escalables a las aplicaciones reales.

Para acceder a los ejemplos pulsando Ejemplos de aplicación en el menú Ayuda de SPSS Modeler. Los archivos de datos y rutas de muestra se instalan en la carpeta *Demos* en el directorio de instalación del producto. [Si desea obtener más información, consulte el tema Carpeta Demos en *Manual de usuario de IBM SPSS Modeler 14.2*.](#)

Ejemplos de modelado de base de datos. Consulte los ejemplos que figuran en el Manual de minería interna de bases de datos de *IBM SPSS Modeler*.

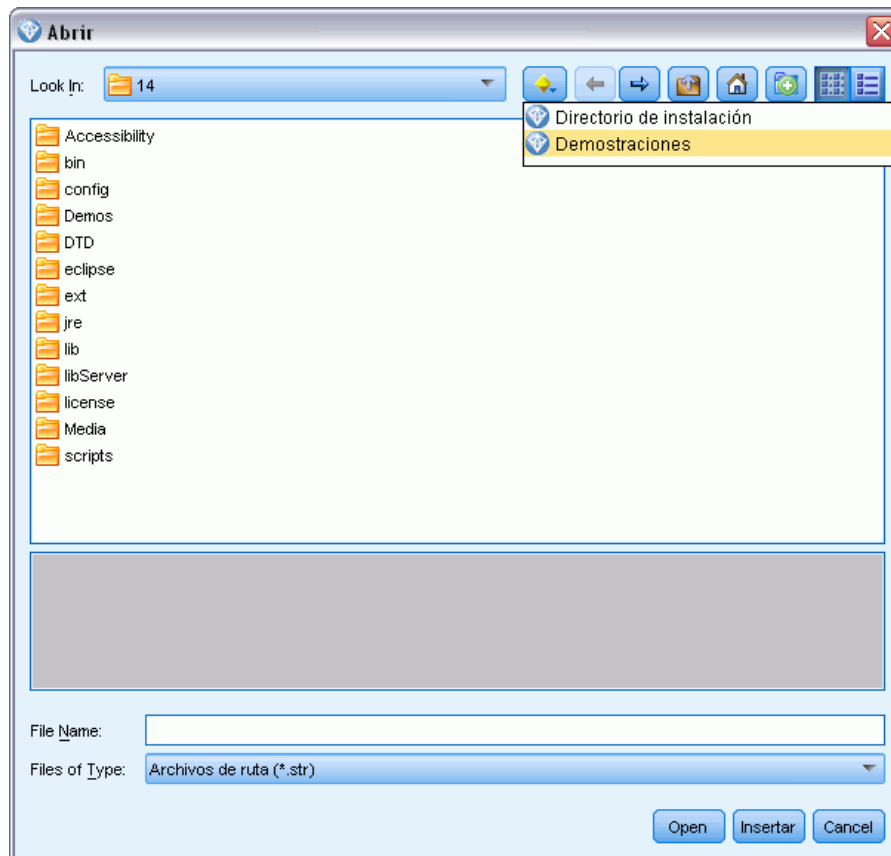
Ejemplos de procesos. Consulte los ejemplos que figuran en la Guía de procesos y automatización de *IBM SPSS Modeler*.

Carpeta Demos

Los archivos de datos y rutas de muestra utilizados con los ejemplos de la aplicación se instalan en la carpeta *Demos* en el directorio de instalación del producto. También puede acceder a esta carpeta desde el grupo de programas IBM SPSS Modeler 14.2 del menú Inicio de Windows o pulsando *Demos* de la lista de directorios recientes en el cuadro de diálogo Abrir archivo.

Figura 1-1

Selección de la carpeta Demos desde la lista de directorios utilizados recientemente



IBM SPSS Modeler Solution Publisher

IBM® SPSS® Modeler Solution Publisher es una potente herramienta de distribución para la integración de resultados de minería de datos en el proceso empresarial para resolver problemas reales. Con SPSS Modeler Solution Publisher, puede crear una versión “empaquetada” de una ruta que se puede ejecutar en un motor de tiempo de ejecución externo o incrustado en una aplicación externa. Esto le permite distribuir las rutas de modelado de datos en un entorno de producción para admitir los procesos empresariales de cada día y proporcionar a las personas encargadas de la toma de decisiones de la organización el conocimiento obtenido a partir de la minería de sus datos.

Utilizar SPSS Modeler Solution Publisher proporciona mayor potencia que simplemente exportar el modelo (como PMML), ya que le permite publicar y distribuir *rutas completas de IBM® SPSS® Modeler*. Esto significa que puede llevar a cabo preparaciones de datos, así como operaciones con registros y campos, como agregar datos, seleccionar registros o derivar nuevos campos antes de crear pronósticos basados en un modelo. De este modo, puede procesar con mayor detalle los resultados de los modelos antes de guardar los datos, y todo simplemente ejecutando la ruta publicada.

Nota: La publicación se ha realizado utilizando la ficha Publicar del nodo de exportación adecuado. [Si desea obtener más información, consulte el tema Publicación de rutas el p. 7](#). Las rutas guardadas en versiones anteriores a la 11.0 del producto empleaban un nodo Publicador separado. El nodo aún se utiliza para la compatibilidad con versiones anteriores, pero ya no está disponible en la paleta del nodo.

Licencia de IBM SPSS Modeler Solution Publisher

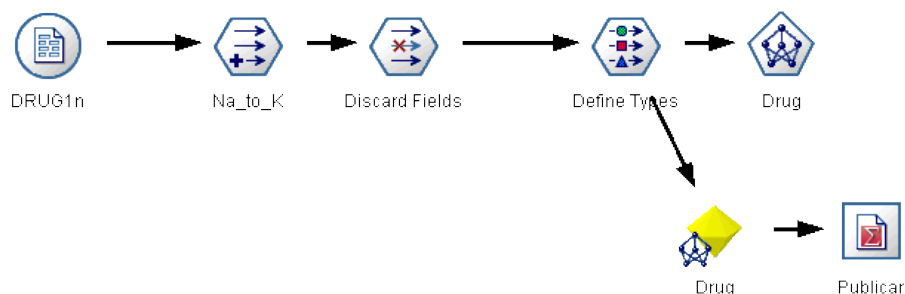
Nota: SPSS Modeler Solution Publisher se distribuye como parte del Servicio de puntuación de IBM® SPSS® Collaboration and Deployment Services, por el cuál se requiere una licencia. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Cuando adquiere una licencia, se proporciona el complemento independiente IBM® SPSS® Modeler Solution Publisher Runtime que permite ejecutar las rutas publicadas. [Si desea obtener más información, consulte el tema Instalación de Runtime en el capítulo 4 el p. 15](#).

Modo de funcionamiento de IBM SPSS Modeler Solution Publisher

La distribución de una solución usando IBM® SPSS® Modeler Solution Publisher consta de dos fases: *publicación* de una ruta y *ejecución* de dicha ruta.

Figura 2-1
Publicación de una ruta



Publicación. Durante el proceso de minería de datos, al final se llega a un modelo que proporciona una buena solución para su problema empresarial. Llegado a ese punto, podrá tomar dicho modelo y aplicarlo a su proceso empresarial. Al publicar una ruta, se guarda en el disco una descripción detallada de la ruta (como un archivo de *imagen* y un archivo de *parámetros*). [Si desea obtener más información, consulte el tema Publicación de rutas el p. 7.](#)

Ejecución. Una vez publicada la ruta, puede volver a crear el proceso implementado en la ruta simplemente ejecutando la ruta publicada. Esto se puede hacer bien usando el ejecutable independiente IBM® SPSS® Modeler Runtime (*modelerrun.exe*) o bien desarrollando una aplicación que utilice la biblioteca de SPSS Modeler Runtime para ejecutar la ruta. Para ejecutar rutas fuera de SPSS Modeler (usando Runtime o una aplicación personalizada), primero debe instalar IBM® SPSS® Modeler Solution Publisher Runtime. [Si desea obtener más información, consulte el tema Instalación de Runtime en el capítulo 4 el p. 15.](#)

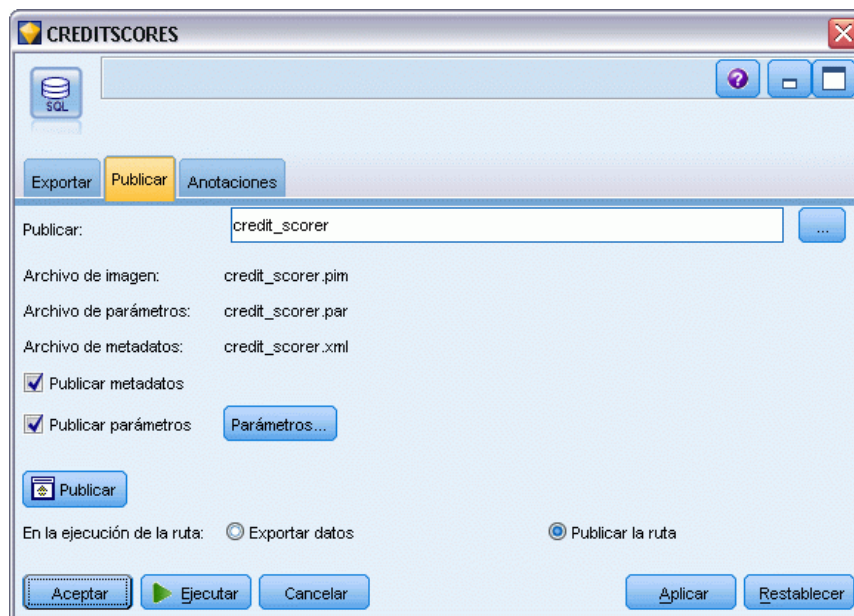
Publicación de rutas

La publicación de rutas se realiza directamente desde IBM® SPSS® Modeler mediante cualquiera de los nodos estándar de exportación: Nodos Base de datos, Archivo plano, Exportar Statistics, Exportar IBM® SPSS® Data Collection, Exportar SAS, Excel y Exportar XML. El tipo de nodo de exportación determina el formato de los resultados que se deben escribir cada vez que se ejecuta la ruta publicada mediante IBM® SPSS® Modeler Solution Publisher Runtime o una aplicación externa. Por ejemplo, si desea escribir los resultados en una base de datos cada vez que se ejecute una ruta publicada, utilice el nodo de exportación Base de datos. [Si desea obtener más información, consulte el tema Conceptos básicos sobre nodos de exportación en el capítulo 7 en *Nodos de origen, proceso y resultado de IBM SPSS Modeler 14.2.*](#)

Para publicar una ruta

- ▶ Abra o cree una ruta de la forma habitual y conecte un nodo de exportación al final.
- ▶ En la ficha Publicar del nodo de exportación, especifique el nombre raíz de los archivos publicados (es decir, el nombre de archivo al que se añadirán las diferentes extensiones *.pim*, *.par* y *.xml*).
- ▶ Pulse en el botón Publicar para publicar la ruta o seleccione Publicar la ruta para publicar la ruta automáticamente cada vez que se ejecute el nodo.

Figura 2-2
ficha Publicar



Nombre publicado. Especifique el nombre raíz de la imagen publicada y los archivos de parámetros.

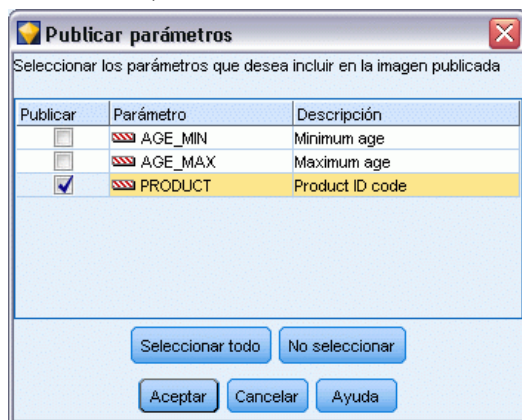
- El **archivo de imagen** (*.pim) ofrece toda la información necesaria para que Runtime ejecute la ruta publicada exactamente tal y como era en el momento de la exportación. Si está seguro de que no necesitará modificar ninguno de los parámetros de configuración de la ruta (como el origen de datos de entrada o el archivo de datos de salida), puede distribuir sólo el archivo de imagen.
- El **archivo de parámetros** (*.par) contiene información configurable sobre orígenes de datos, archivos de resultados y opciones de ejecución. Si desea poder controlar la entrada o la salida de la ruta sin volver a publicarla, deberá publicar el archivo de parámetros además del archivo de imagen.
- El **archivo de metadatos** (*.xml) describe las entradas y salidas de la imagen y sus modelos de datos. Está diseñado para que lo utilicen aplicaciones que incrustan la biblioteca de Runtime y que necesitan conocer la estructura de los datos de entrada y salida.

Nota: Este archivo sólo se produce si selecciona el cuadro Publish metadata (Publicar metadatos).

Publish parameters (Parámetros publicados). Si es necesario, puede incluir parámetros de ruta en el archivo *.par. Puede cambiar estos valores de parámetros de ruta cuando ejecute la imagen editando el archivo *.par o a través de la API de tiempo de ejecución.

Esta casilla de verificación activa el botón Parámetros. Cuando pulsa en el botón, aparece el cuadro de diálogo Publish Parameters (Parámetros publicados).

Figura 2-3
Selección de parámetros



Seleccione los parámetros que desee incluir en la imagen publicada seleccionando la casilla de verificación relevante en la columna Publicar.

En la ejecución de la ruta. Especifica si la ruta se publicará automáticamente cuando se ejecute el nodo.

- **Exportar datos.** Ejecuta el nodo de exportación de la forma habitual, sin publicar la ruta. (Básicamente el nodo se ejecuta en SPSS Modeler como si IBM® SPSS® Modeler Solution Publisher no estuviera disponible.) Si selecciona esta opción, la ruta no se publicará a no ser que lo haga expresamente pulsando en el botón Publicar en el cuadro de diálogo del nodo de exportación. Si lo prefiere, puede publicar la ruta actual mediante la herramienta Publicar de la barra de herramientas o mediante un proceso.
- **Publicar la ruta.** Publica la ruta para su distribución con SPSS Modeler Solution Publisher. Seleccione esta opción si desea publicar de forma automática la ruta cada vez que se ejecute.

Comentarios

- Si pretende ejecutar la ruta publicada con datos nuevos o actualizados, es importante que el orden de los campos del archivo de entrada sea el mismo que el de los campos del archivo de entrada del nodo de origen especificado en la ruta publicada.
- Cuando se publican datos en aplicaciones externas, es recomendable filtrar los campos no pertinentes o cambiar el nombre de los campos para que cumplan los requisitos de entrada. Ambas acciones se pueden realizar mediante un nodo Filtrar antes del nodo de exportación. [Si desea obtener más información, consulte el tema Cambio de nombre o filtrado de campos en el capítulo 4 en *Nodos de origen, proceso y resultado de IBM SPSS Modeler 14.2*.](#)

Ejecución de rutas publicadas

El principal efecto de la ejecución de una ruta publicada es siempre la generación de un conjunto de datos. Los datos se pueden guardar en un archivo de disco o en una base de datos. Las rutas publicadas no pueden generar gráficos, tablas, modelos generados u otro resultado que no sea datos.

Las rutas publicadas con IBM® SPSS® Modeler Solution Publisher se pueden ejecutar con el programa IBM® SPSS® Modeler Solution Publisher Runtime. El programa Runtime, *modelerrun.exe*, se inicia desde la línea de comando, con opciones que indican la ejecución del archivo de imagen y (opcionalmente) el archivo de parámetros. El comando de Runtime es el siguiente:

```
modelerrun -nobanner -p <archivo de parámetros> -o <opciones> <archivo de imagen>
```

donde

-nobanner	suprime el mensaje de texto de inicio de Runtime (opcional).
<archivo de parámetros>	es el nombre del archivo de parámetros (opcional).
<opciones>	es una lista de opciones separadas por comas que constituyen pares opción=valor y especifican las opciones de ejecución de Runtime. Más adelante se detallan las opciones (opcional).
<archivo de imagen>	es el nombre del archivo de imagen que ejecutar (obligatorio).

Están disponibles las siguientes opciones de ejecución:

Tabla 3-1
Opciones válidas de Runtime

Name	Valor por defecto	Descripción
max_file_size	-1	Tamaño máximo de los archivos guardados. El valor -1 indica que no hay límite.
max_sql_string_length	2,048	Longitud máxima de una cadena importada de una base de datos con SQL. Los valores de cadena con más caracteres se truncan por la derecha sin advertencias. El rango válido es de 1 a 65.535 caracteres.
memory_usage	100	Multiplicador de asignación de la memoria dinámica. Ajuste este valor para regular el uso total de memoria del servidor.
temp_directory	"	Especifica el directorio que utilizar para los archivos temporales.
request_passwords	Y	Controla si se debe pedir una contraseña de base de datos cuando se ejecute una ruta publicada que requiera acceso a una base de datos (Y/N).

Uso del archivo de parámetros para personalizar la ejecución de una ruta

Cuando se publica una ruta, el archivo de imagen contiene información sobre el origen de datos y el destino de los datos de salida seleccionados en la ruta original. No obstante, a veces resulta conveniente ejecutar una ruta con datos de distinta procedencia o redirigir los datos de salida a un destino diferente. Esto se puede hacer editando el archivo de parámetros de la ruta.

Se trata de una tarea de especial importancia cuando se ejecuta la ruta publicada en un ordenador diferente al que se utilizó para crearla. Deberá actualizar las ubicaciones de los archivos de entrada y salida para reflejar la estructura del ordenador de destino. Tenga en cuenta que las rutas publicadas no admiten rutas relativas, lo que da una relevancia especial a la verificación de las ubicaciones de los datos en el archivo de parámetros. (Si especifica rutas relativas en las opciones del nodo de exportación al publicar la ruta, IBM® SPSS® Modeler las convertirá en rutas absolutas antes de crear los archivos de imagen y de parámetros de la ruta publicada.)

También puede utilizar el archivo de parámetros para definir las opciones de ejecución y cambiar los valores de los parámetros de ruta.

Los valores de los parámetros se especifican en el archivo de parámetros como pares <nombre. atributo>=<valor>. En el caso de algunos parámetros, el nombre del parámetro contiene un dígito identificativo para distinguir los nodos del mismo tipo en la ruta. Por ejemplo, en una ruta donde la entrada y la salida hagan referencia a una conexión con una base de datos, los parámetros que describen la conexión de entrada podrían ser `dbconn0.datasource`, `dbconn0.user`, etc., y los asociados con la conexión de salida serían `dbconn1.datasource`, `dbconn1.user`, etc. Los valores siempre se acotan con comillas dobles. El archivo de parámetros contiene los siguientes parámetros, que se pueden modificar para personalizar la ejecución de la ruta.

Tabla 3-2
Parámetros de ejecución

Parámetro	Propósito
<code>dbconnx.datasource</code>	Nombre de origen de datos (DSN).
<code>dbconnx.user</code>	Nombre del usuario para bases de datos con acceso restringido.
<code>dbconnx.password</code>	Contraseña para bases de datos con acceso restringido.
<code>dbconnx.epassword</code>	Contraseña codificada para bases de datos con acceso restringido. Para generar una contraseña codificada, seleccione Codificar contraseña en el menú Herramientas de la interfaz de usuario de SPSS Modeler. Copie y pegue la contraseña codificada si es necesario.
<code>dbconnx.password_required</code>	Marca que indica si se debe solicitar una contraseña para la conexión con esta base de datos. (Los valores son "Y" y "N".) Se establece automáticamente en "N" cuando no se especifica ninguna contraseña en el momento de la publicación.
<code>dbconnx.catalog</code>	Catálogo asociado con la conexión a la base de datos.
<code>dbtablex.name</code>	Nombre de tabla de base. <i>Nota:</i> el orden de los campos de una nueva tabla debe coincidir con el de los campos de la ruta originalmente publicada.
<code>dbtablex.schema</code>	Nombre del esquema.
<code>dbtablex.catalog</code>	Catálogo asociado con la base de datos actual.
<code>filex.name</code>	Nombre del archivo de datos. <i>Nota:</i> el orden de los campos de un nuevo archivo de entrada debe ser el mismo que el orden de los campos de la ruta originalmente publicada.

Parámetro	Propósito
filex.path	Ubicación del archivo (nombre del directorio)
filex.field_names_included	En el caso de archivos con formato de texto, es la marca que indica si los nombres de los campos se incluyen como primera línea en el archivo. (Los valores son “Y” y “N”.)
filex.field_separator	En el caso de archivos con formato de texto, son los caracteres utilizados para separar los valores de los campos.
filex.decimal_separator	En el caso de archivos con formato de texto, es el carácter utilizado para indicar el separador decimal.
options.angle_in_radians	Marca que indica si se deben usar radianes como unidad de medida en las expresiones trigonométricas de CLEM. (Los valores son “Y” y “N”.)
options.date_2digit_baseline	Define el siglo en las fechas especificadas con años de dos dígitos.
options.date_baseline	El año de línea base (siempre 1 de enero) que utilizarán las funciones de fecha de CLEM que trabajan con una sola fecha.
options.time_rollover	Marca que indica si las diferencias de tiempo negativas hacen referencia al pasado.
options.decimal_separator	Carácter por defecto para el separador decimal utilizado en los archivos de texto.
options.time_format	Formato de hora que se debe usar cuando las funciones de hora de CLEM interpretan las cadenas como horas.
options.date_format	Formato de fecha que se debe usar cuando las funciones de fecha de CLEM interpretan las cadenas como fechas.
options.timestamp_format	Formato utilizado para leer los campos de marca de tiempo como cadenas desde los orígenes de datos ODBC.
paramx.value	Valor del parámetro de ruta. <i>Nota:</i> El nombre del parámetro, descripción y tipo de almacenamiento se incluyen como referencia pero no se pueden cambiar. El valor tiene que ser válido para el tipo.

Incrustación de IBM SPSS Modeler Solution Publisher en las aplicaciones

Además del motor independiente de Runtime, IBM® SPSS® Modeler Solution Publisher ofrece una biblioteca de programación de tiempo de ejecución (CLEMRTL) que permite a otros programas controlar la ejecución de rutas de SPSS Modeler Solution Publisher. Puede llamar a procedimientos de CLEMRTL desde programas cliente desarrollados en C, C++. Para usar CLEMRTL, deberá incluir el archivo de encabezado *clmrtl.h* (disponible en la carpeta *..\installation\clmrtl\include*) y asegurarse de vincular el archivo de biblioteca adecuado para su plataforma de desarrollo en la aplicación cuando lo genere. El archivo *clmrtl.dll* está disponible en la carpeta *..\installation\bin*.

Cualquier archivo de origen que haga referencia a procedimientos de biblioteca debe incluir el archivo de encabezado *clmrtl.h*. Este archivo de encabezado proporciona prototipos ANSI C para los procedimientos de biblioteca y define macros de gran utilidad. No requiere incluir otros encabezados aparte de los que requiera el programa. Para evitar los conflictos de nombres, todos los nombres de funciones y tipos de biblioteca comienzan por *clmrtl_* y todos los nombres de macro contienen el prefijo *CLEMRTL_*.

CLEMRTL produce informes con información útil que se debe comunicar de nuevo a la aplicación de algún modo. CLEMRTL ofrece tres mecanismos para gestionar dichos mensajes:

- Los mensajes se pueden guardar en un archivo de registro especificado por la función `clemrtl_setLogFile()`.
- La aplicación puede recuperar los detalles del último error usando la función `clemrtl_getErrorDetail()`.
- La aplicación puede ofrecer su propio procedimiento de gestión de informes usando la función `clemrtl_setReportHandler()`.

El sistema de Runtime contiene un catálogo de mensajes localizados para informes. Cualquier informe que se pase a la aplicación debería incluir la cadena del mensaje localizado. Los informes también incluyen el código del informe para que la aplicación pueda elegir la interpretación y presentación del mensaje de maneras diferentes. Con el fin de obtener mensajes detallados (localizados), la aplicación debe albergar el archivo apropiado `messages.cfg` en un directorio `config`. Podrá encontrar distintas versiones de `messages.cfg` para las diferentes configuraciones regionales en el directorio `<DirInstalación>/config/<configuración regional>`, donde `<DirInstalación>` es el directorio en el que ha instalado SPSS Modeler Solution Publisher y `<configuración regional>` es la configuración regional deseada. Seleccione la versión apropiada de `messages.cfg` y añádala al subdirectorio `config` del directorio del proyecto.

El sistema de Runtime no necesita ninguna configuración especial de entorno ni registro para funcionar correctamente. Se deben distribuir bibliotecas independientes con una aplicación vinculada a CLEMRTL. Los archivos de ejemplo se instalan en el directorio `<DirInstalación>/bin`.

Requisitos

A continuación se detallan las plataformas y los compiladores verificados con la biblioteca de programación de Runtime:

Tabla 3-3
Plataformas y compiladores

Plataforma	Compiladores recomendados
AIX	IBM XL C/C++ Enterprise Edition V8.0 para AIX
HP-UX 11.0	aC++/ANSI C B3910B A.06.05
Solaris 64 bits	Sun Studio 11 5.8 revisión 121017-06
Linux 32 bits y 64 bits	gcc 3.2.3, Intel C++ Compiler 9.1.043
Windows	Visual Studio 2005

Nota: En HP-UX, la aplicación se debe crear con la marca “-AA” del compilador aCC para permitir la vinculación con las bibliotecas CLEMRTL.

Aplicación de muestra de IBM SPSS Modeler Solution Publisher

Se incluye una aplicación de ejemplo, `dlltest.c`, en la carpeta `\clemrtl\demo` del directorio de instalación de IBM® SPSS® Modeler Solution Publisher. Use la aplicación de ejemplo para iniciarse en la incrustación de la biblioteca de Runtime en sus propias aplicaciones.

La información utilizada por la aplicación de ejemplo se incluye en la carpeta `\clemtr\demo`; consulte el archivo `LEAME.txt` para obtener instrucciones. La compilación de una aplicación difiere de una plataforma a otra, por lo que también se incluyen archivos de creación de ejemplo.

Instalación de IBM SPSS Modeler Solution Publisher Runtime

Instalación de Runtime

Cuando adquiere IBM® SPSS® Modeler Solution Publisher como un componente complementario para IBM® SPSS® Modeler, recibe un disco de instalación independiente. Utilice este disco para instalar IBM® SPSS® Modeler Solution Publisher Runtime, que permite ejecutar las rutas publicadas. Las rutas publicadas son independientes del sistema operativo, así que se puede ejecutar cualquier ruta publicada en cualquier plataforma que Runtime admita.

Nota: la funcionalidad que permite publicar rutas desde SPSS Modeler viene instalada con IBM® SPSS® Modeler y se puede activar comprando un código de autorización independiente. Tenga en cuenta que los archivos publicados deben volver a exportarse con cada versión de SPSS Modeler. Por ejemplo, un archivo exportado con una versión anterior no funcionará con la versión 13 de SPSS Modeler Solution Publisher Runtime.

Requisitos del sistema

Los requisitos del sistema para instalar IBM® SPSS® Modeler Solution Publisher Runtime son:

- **Sistema operativo.** Windows Server 2008 R2 para sistemas x64 de 64 bits; Windows Server 2003 Standard Edition R2 para sistemas x86 de 32 bits o x64 de 64 bits
IBM AIX® 6.1 o 7.1 para sistemas POWER de 64 bits; HP-UX 11i V3 para sistemas Itanium de 64 bits; Oracle Solaris™ 9.x o 10 para sistemas SPARC de 64 bits; Red Hat Enterprise Linux 5.x para sistemas x86 de 32 bits; Red Hat Enterprise Linux 5.x o 6.x para x64 de 64 bits o IBM System z; Red Hat Enterprise Linux Advanced Platform 5.x para sistemas x86 de 32 bits; Red Hat Enterprise Linux Advanced Platform 5.x o 6.x para sistemas x64 de 64 bits o IBM System z; SuSE Linux Enterprise Server 10 o 11 para sistemas x64 de 64 bits o IBM System z
- **Hardware.** Procesador Pentium® o Pentium-class or superior (para Windows de 32 bits); familia de procesadores x64 (AMD 64 y EM64T) (para Windows de 64 bits), procesador PowerPC ejecutando a 1 GHz o más
, 233 MHz o más e IBM System p para IBM AIX; procesador Itanium para HP-UX; UltraSPARC II (o superior) para Solaris; procesador Pentium o Pentium-class o superior para Linux de 32 bits; familia de procesadores x64 (AMD 64 y EM64T) o IBM System z para Linux de 64 bits
- **Cantidad mínima de espacio libre en disco.** Se recomienda 1 gigabyte (GB). Se recomienda 2 GB si se ejecuta en HP-UX.
- **RAM.** Se recomiendan 4 GB.
- **Entorno virtual.** Los siguientes entornos virtuales admiten IBM® SPSS® Modeler Server.
- IBM PowerVM Hypervisor para IBM AIX

- IBM PR/SM o z/VM para Linux para IBM System z
- VMWare ESX Server 4.1
- VMWare vSphere 4.0
- **Entorno de conglomerado.** Los siguientes entornos de conglomerado admiten SPSS Modeler Server.
 - IBM HACMP/RSCT para IBM AIX
 - IBM WebSphere App Server de 64 bits
 - Microsoft Cluster Service para Windows
 - Oracle Solaris Cluster
 - Oracle WebLogic App Server de 64 bits
 - Red Hat Cluster Suite para Red Hat Enterprise Linux

Instalación en Windows

La siguiente sección describe el procedimiento de instalación de SPSS Modeler Solution Publisher Runtime en el sistema operativo Windows.

- ▶ Inserte el disco de instalación IBM SPSS Modeler Solution Publisher en la unidad.
- ▶ En el menú de instalación, seleccione Instalar IBM® SPSS® Modeler Solution Publisher Runtime o Instalar IBM® SPSS® Modeler Solution Publisher Runtime para Win64, dependiendo del sistema al que está realizando la instalación.
- ▶ En el asistente de instalación, pulse en Siguiente para empezar.
- ▶ Siga las instrucciones que aparecen en la pantalla. Para continuar, pulse en Siguiente.
- ▶ Una vez que haya especificado todas las opciones, podrá instalarlo. Pulse en Instalar para comenzar la transferencia de archivos.
- ▶ Después de instalar todos los archivos, pulse en Finalizar.

Instalación en UNIX

Las secciones siguientes describen los requisitos adicionales del sistema y el procedimiento de instalación de IBM® SPSS® Modeler Solution Publisher Runtime en el sistema operativo UNIX.

Requisitos adicionales para UNIX

Debe asegurarse de que los límites de kernel del sistema son suficientes para el funcionamiento de IBM® SPSS® Modeler Server. Los límites de datos, memoria y archivos son especialmente importantes y deben establecerse como ilimitados dentro del entorno de SPSS Modeler Server. Para ello:

- ▶ Añada los siguientes comandos a *modelersrv.sh*:

```
ulimit -d unlimited
```

```
ulimit -m unlimited
```

```
ulimit -f unlimited
```

- ▶ Reinicie SPSS Modeler Server.

Solaris.

Además de los requisitos del sistema anteriormente mencionados, la instalación en Solaris también requiere las siguientes revisiones específicas de la versión. Las revisiones se pueden descargar de <http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage>.

- *Identificadores de revisiones para Solaris 2.9*: 111711-16, 111712-16 y 111722-05 (C++)
- *Identificadores de revisiones para Solaris 2.10*: 119963 (C++)

La biblioteca */usr/lib/iconv/646%UTF-8* también debe estar instalada en el sistema. Si actualmente no está instalada, instale el paquete *SUNWiniu8* o *SUNWiniu8x*, que instalará dicha biblioteca como parte de los módulos Indic iconv para UTF-8.

Linux de 32 bits.

Para poder ejecutarlo correctamente son necesarios los siguientes paquetes adicionales. Estos paquetes se pueden descargar de Red Hat Network. Es posible que, para acceder la red, deba registrarse (<https://rhn.redhat.com/network/software/search.pxt>). Utilice la función de búsqueda para buscar y profundizar en las versiones específicas de los paquetes.

- El paquete *libstdc++* (versión: *libstdc++-3.2.3-X.Y.i386* donde *X.Y* es 22.1 o posterior).
- El paquete *libgcc* (versión: *libgcc-3.2.3-X.Y.i386* donde *X.Y* es 22.1 o posterior).
- El paquete *compat-libstdc++* (versión: *compat-libstdc++-33-3.2.3-X.Y.i386* donde *X.Y* es 47.3 o posterior).

Linux de 64 bits.

Para poder ejecutarlo correctamente son necesarios los siguientes paquetes adicionales. Estos paquetes se pueden descargar de Red Hat Network. Es posible que, para acceder la red, deba registrarse (<https://rhn.redhat.com/network/software/search.pxt>). Utilice la función de búsqueda para buscar y profundizar en las versiones específicas de los paquetes.

- El paquete *libstdc++* (versión: *libstdc++-3.2.3-X.Y.x86_64* donde *X.Y* es 22.1 o posterior).
- El paquete *libgcc* (versión: *libgcc-3.2.3-X.Y.x86_64* donde *X.Y* es 22.1 o posterior).
- El paquete *compat-libstdc++* (versión: *compat-libstdc++-33-3.2.3-X.Y.x86_64* donde *X.Y* es 47.3 o posterior).

AIX.

Además de los requisitos del sistema anteriormente mencionados, la instalación en AIX también requiere lo siguiente:

- El tiempo de ejecución de XL C++ debe ser al menos de nivel 8.0.0.9 (lspp -l xIC.rte). Los conjuntos de datos que contienen las bibliotecas se denomina IBM C++ Runtime Environment Components for AIX y se pueden descargar desde <http://www.ibm.com/software/awdtools/xlcpp/support/>.
- El tiempo de ejecución de XL FORTRAN debe ser al menos de nivel 10.1.0.4 (lspp -l xlfite). Los conjuntos de datos que contienen las bibliotecas se denomina XL Fortran Enterprise Edition V10.1 for AIX Runtime Environment y se pueden descargar desde <http://www.ibm.com/software/awdtools/fortran/xlfortran/support/>.

HP-UX.

Además de los requisitos del sistema anteriormente mencionados, la instalación en HP-UX también requiere que se instalen las bibliotecas de tiempo de ejecución de C++. Puede descargarlas como la revisión PHSS_26946, tal y como se muestra en la siguiente tabla. Para acceder a las revisiones, vaya a <http://www4.itrc.hp.com/service/patch/mainPage.do>. Aunque es necesario registrarse, es un servicio completamente gratuito.

Procedimiento de instalación de UNIX

Estas instrucciones de instalación se aplican a IBM® SPSS® Modeler Solution Publisher Runtime para UNIX. Necesita permisos de escritura y lectura para el directorio de instalación de destino, por lo que debe iniciar sesión con una cuenta que tenga permisos suficientes.

- ▶ Inserte el disco de instalación en la unidad y asegúrese de que puede acceder a la unidad. En Solaris, debe estar ejecutándose el administrador de volúmenes (el demonio vold).
- ▶ Cambie el directorio al punto de montaje del DVD.
- ▶ Cambie los directorios al directorio *modelrun*.
- ▶ Cambie al directorio de la plataforma pertinente.
- ▶ Ejecute el proceso de instalación de *.bin*. Por ejemplo:

```
./modelersolutionpublisherhpia64.bin -i console
```
- ▶ Se muestran los detalles de introducción. Pulse Intro para continuar.
- ▶ Se mostrará la información de licencias. Lea la licencia, escriba 1 para aceptarla y pulse Intro para continuar.
- ▶ Se muestra una lista de los idiomas disponibles. Introduzca el número de la versión del idioma que desea instalar y pulse Intro para continuar.
- ▶ Se le solicitará que escriba la ubicación de instalación. Para usar el directorio por defecto, */usr/IBM/SPSS/ModelerSolutionPublisher<nn>* (donde *<nn>* es el número de versión), pulse Entrar. Se instalará Runtime en el directorio especificado.

- ▶ Se le solicitará que confirme la ubicación de instalación. Cuando sea correcta, escriba y pulse Intro.
- ▶ Se mostrará un resumen de instalación previa para confirmar sus entradas hasta el momento. Pulse Intro para continuar.
- ▶ Se mostrará un mensaje que le comunicará que la rutina de instalación está lista para ejecutarse. Pulse Intro para continuar.
- ▶ Se mostrará una barra de progreso durante la ejecución de la rutina de instalación. Cuando finalice la instalación, pulse Intro para salir del instalador.

Inicio de Runtime

Una vez que se ha instalado Runtime, se puede utilizar para ejecutar rutas que se hayan publicado desde IBM® SPSS® Modeler utilizando cualquiera de los nodos de exportación. [Si desea obtener más información, consulte el tema Publicación de rutas en el capítulo 2 el p. 7.](#) Inicie Runtime desde la línea de comandos, con opciones que indiquen el nombre del archivo de imagen publicado y el archivo de parámetro opcional que se va a ejecutar.

Para ejecutar Runtime, escriba la siguiente línea en el símbolo del sistema:

```
modelerrun -p <archivo de parámetros> -o <opciones> <archivo de imagen>
```

donde

<parameter-file>	es el nombre del archivo del archivo de parámetros publicado (opcional).
<options>	es una lista de opciones separadas por comas que constituyen pares option=value y especifican las opciones de ejecución de Runtime.
<image file>	es el nombre del archivo de imagen publicado (*.pim) que se va a ejecutar.

Nota: si utiliza lenguajes multibyte como chino o japonés en un entorno UNIX, debe especificar los códigos de lenguaje que se van a usar. Los siguientes ejemplos muestran los códigos de lenguaje de chino simplificado:

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

donde -p es la ubicación de los archivos pim y par.

[Si desea obtener más información, consulte el tema Ejecución de rutas publicadas en el capítulo 3 el p. 10.](#)

Modificación del directorio temporal

IBM® SPSS® Modeler Solution Publisher Runtime realiza algunas operaciones que requieren la creación de archivos temporales. Por defecto, Runtime utiliza el directorio temporal del sistema para crear archivos temporales. Se puede modificar la ubicación del directorio temporal con los pasos siguientes.

Nota: Es muy importante utilizar la sintaxis correcta, siendo especialmente cuidadoso con los espacios y la ruta del directorio.

Suponga que está utilizando SPSS Modeler Solution Publisher Runtime con los siguientes archivos publicados: *webtest.pim* y *webtest.par*. En esta situación, desea modificar la ubicación de los archivos temporales para utilizar el directorio *C:\published\temp*. Para cambiar la ubicación del directorio temporal utilizado por el ejecutable:

- ▶ Desde el directorio donde están guardados los archivos *.pim* y *.par*, ejecute el comando:

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

De esta forma se obliga a Runtime a usar “*C:\published\temp*” como directorio temporal sólo para ese proceso específico.

Desinstalación de IBM SPSS Modeler Solution Publisher Runtime

La sección siguiente describe los procedimientos para la desinstalación de IBM® SPSS® Modeler Solution Publisher Runtime.

Procedimiento de desinstalación para Windows

- ▶ En el Panel de control de Windows, abra Agregar o quitar programas.
- ▶ Seleccione IBM® SPSS® Modeler Solution Publisher <nn> en la lista que aparece.
- ▶ Pulse en Eliminar.
- ▶ Siga las instrucciones que aparecerán en la pantalla para desinstalar Runtime.

Nota: para quitar programas, debe haber iniciado sesión en el ordenador con privilegios de administrador.

Procedimiento de desinstalación para UNIX

- ▶ Quite el directorio de instalación, incluidos todos los archivos instalados.

Referencia API de la biblioteca de IBM SPSS Modeler Solution Publisher Runtime

Introducción

La API de CLEMRTL permite controlar la ejecución de las rutas publicadas desde su aplicación. Esta sección describe las funciones API disponibles.

A continuación se muestran algunos problemas generales al utilizar API:

- La API tiene vinculación C para una máxima compatibilidad, pero la biblioteca tiene dependencias C++. En algunas plataformas, esto podría significar que sólo se puede utilizar con un vinculador de C++.
- Se debe llamar `clemrtl_initialise_ext()` antes de utilizar cualquier otra función de la biblioteca.
- El tipo `clemrtl_image_handle` se utiliza como un identificador de imagen.
- Cada función devuelve un código de indicador de estado. Los valores del indicador de estado son:

CLEMRTL_OK	Éxito
CLEMRTL_FAIL	Error sin más detalles disponibles
CLEMRTL_ERROR	Error con información adicional sobre el error disponible a través de la función <code>clemrtl_getErrorDetail()</code>

Conceptos básicos del proceso de API

El esquema general de una aplicación que utiliza API es el siguiente:

- ▶ Inicializar la biblioteca mediante `clemrtl_initialise_ext()`.
- ▶ Comprobar y cambiar las opciones de ejecución mediante `clemrtl_getOption()` y `clemrtl_setOption()`.
- ▶ Abrir una imagen mediante `clemrtl_openImage()` y recibir un controlador de imagen.
- ▶ Comprobar y cambiar los parámetros de la imagen mediante `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` y `clemrtl_setParameter()`.
- ▶ Ejecutar la imagen utilizando `clemrtl_execute()`. Si se va a ejecutar varias veces la misma imagen sin cambiar los parámetros, se utiliza primero `clemrtl_prepare()`.
- ▶ Cerrar la imagen mediante `clemrtl_closeImage()`.

- ▶ Para cancelar la ejecución en curso, se utiliza `clemrtl_interrupt()`.
- ▶ Para recuperar información sobre el último error, se utiliza `clemrtl_getErrorDetail()`. Para recibir mensajes de error y de diagnóstico según lleguen, se utiliza `clemrtl_setReportHandler()`.

La biblioteca está diseñada para utilizarla en subprocesamiento múltiple bajo las condiciones siguientes:

- No se puede llamar de manera simultánea a funciones globales, que no toman un controlador de imagen como un argumento.
- No se puede llamar de manera simultánea a funciones locales de imagen, que toman un controlador de imagen como un argumento, cuando se aplican al mismo controlador de imagen, a excepción de `clemrtl_interrupt()`.
- Se puede llamar de manera simultánea a funciones locales de imagen cuando se aplican a diferentes controladores de imagen.

Concretamente, esto significa que varios subprocesos distintos pueden preparar y ejecutar imágenes de manera simultánea, siempre que utilicen diferentes controladores de imagen. Es posible que varios controladores distintos hagan referencia a la misma imagen (`clemrtl_openImage()` crea un nuevo controlador cada vez que se le llama, incluso para la misma imagen). No obstante, cuando ejecute varias instancias de la misma imagen, asegúrese de cambiar los parámetros de la imagen para volver a direccionar los resultados según sea necesario. Si dos imágenes que se están ejecutando de manera simultánea intentan escribir en el mismo archivo de resultados o tabla de base de datos, los resultados serán impredecibles.

Funciones API

A continuación se muestra una lista completa de las funciones expuestas en la API.

initialise

Esta función está desaprobadada y es equivalente a

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

Los nuevos programas deben llamar *initialise_ext*.

initialise_ext

```
int clemrtl_initialise_ext(  
    marcas sin signo,  
    int arg_count,  
    const clemrtl_init_arg* args);
```

Inicializa IBM® SPSS® Modeler Solution Publisher Runtime. *Debe llamarse a esta función antes que a las otras funciones API.*

Parámetro	Descripción
marcas	Controla algunos aspectos del proceso de inicialización. El valor está constituido como OR a nivel de bit de las marcas descritas a continuación.
arg_count	El número de argumentos de inicialización adicionales. Debe ser igual o inferior a la longitud de la matriz args.
args	Argumentos de inicialización adicionales. Los argumentos se describen según el tipo: <pre>typedef struct _clemrtl_init_arg { const char* nombre; const char* valor; } clemrtl_init_arg;</pre>

El campo nombre es el nombre de una propiedad de configuración y el campo valor es su valor. Se reconocen las siguientes propiedades de configuración:

Nombre	Valor
directorio_instalación	La ruta completa a la carpeta de instalación de IBM® SPSS® Modeler Solution Publisher. La carpeta de instalación es la que contiene las carpetas bin y config.

Devuelve uno de los siguientes códigos de error:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error al inicializar, sin más detalles disponibles.

El parámetro marcas proporciona control sobre ciertos aspectos del proceso de inicialización. Un valor de 0 (o CLEMRTL_INIT_DEFAULTS) especifica el comportamiento por defecto como se indica a continuación:

- Establezca la zona horaria por defecto, llamando a tzset().
- Establezca la configuración regional LC_NUMERIC en "C".
- Establezca el nuevo controlador global para iniciar una excepción cuando la memoria esté vacía.

Este comportamiento se puede modificar especificando combinaciones de las siguientes marcas:

Marcar	Descripción
CLEMRTL_INIT_NO_TZ	No establezca la zona horaria.
CLEMRTL_INIT_NO_LOCALE	No cambie la configuración regional.
CLEMRTL_INIT_NO_NEW_HANDLER	No establezca el controlador nuevo.
CLEMRTL_INIT_LOCAL_NEW_HANDLER	Localice el nuevo controlador en cada llamada API.

La biblioteca requiere que la configuración regional `LC_NUMERIC` se establezca en “C” para convertir correctamente entre números y cadenas (el formato numérico de IBM® SPSS® Modeler utiliza las reglas de la configuración regional de “C”). Esto incluye la conversión de números leídos y escritos en archivos de texto.

- Especifique `NO_LOCALE` si su aplicación depende de la configuración regional `LC_NUMERIC` y no requiere conversión entre números y cadenas.
- `NO_NEW_HANDLER` y `LOCAL_NEW_HANDLER` se excluyen entre sí. La biblioteca requiere que cuando la memoria está vacía, `new` debe iniciar una excepción.
- Especifique `NO_NEW_HANDLER` si la aplicación establece su propio controlador nuevo o si está seguro de que el compilador/sistema de Runtime inicia una excepción por defecto.

Especifique `LOCAL_NEW_HANDLER` sólo si su aplicación depende de que `new` devuelva un puntero nulo cuando la memoria está vacía. *Nota:* Esta operación no es segura en un entorno de subprocesamiento múltiple.

getOption

```
int clemrtl_getOption(
    const char* nombre,
    char* valor,
    int tamaño_valor);
```

Recupera el valor de una opción de ejecución.

Parámetro	Descripción
nombre	El nombre de la opción.
valor	Un búfer para recibir el valor de la opción.
tamaño_valor	El tamaño del búfer del valor.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el nombre de la opción no es ninguno de los nombres de opción de ejecución de la lista anterior (por ejemplo, `memory_usage`) o si el búfer del valor no es lo suficientemente grande como para recibir el valor que incluye un carácter nulo en finalización. Los valores de opción son siempre cadenas, incluso cuando la interpretación es numérica. [Si desea obtener más información, consulte el tema Ejecución de rutas publicadas en el capítulo 3 el p. 10.](#)

setOption

```
int clemrtl_setOption(
    const char* nombre,
    const char* valor);
```

Define el valor de una opción de ejecución.

Parámetro	Descripción
nombre	El nombre de la opción.
valor	El valor de la opción.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el nombre de la opción no es ninguno de los nombres de opción de ejecución de la lista anterior (por ejemplo, `memory_usage`). Los valores de opción son siempre cadenas, incluso cuando la interpretación es numérica.

setLogFile

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

Redirige los mensajes de registro desde todas las imágenes abiertas posteriormente.

Parámetro	Descripción
moda	El modo de registro debe ser uno de los valores siguientes: CLEMRTL_NULL_LOG : suprime mensajes. CLEMRTL_STDERROR_LOG : envía mensajes al resultado de error estándar. CLEMRTL_FILE_LOG : envía mensajes al archivo cuyo nombre ha asignado el parámetro <code>archivo_registro</code> .
archivo_registro	Cadena que contiene el nombre de archivo.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el modo no es ninguno de los modos reconocidos o si el registro del archivo se solicita y el nombre de archivo no se ha especificado o no es válido.

openImage

```
int clemrtl_openImage(
    const char* archivo_imagen,
    const char* archivo_parámetro,
    clemrtl_image_handle* controlador);
```

Abre una imagen de una ruta publicada.

Parámetro	Descripción
archivo_imagen	El nombre de archivo de la imagen.
archivo_parámetro	El nombre de archivo del parámetro.
identificador	Recibe el controlador de la imagen.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

El archivo del parámetro puede ser NULL si no se requiere un archivo de parámetro distinto. La operación da error si el archivo de imagen no se especifica, el nombre de archivo no es válido o el contenido no se puede leer. Si se completa la operación, el controlador de imagen devuelto se puede utilizar para identificar la instancia de la imagen en las llamadas API posteriores.

closelImage

```
int clemrtl_closelImage(
    clemrtl_image_handle controlador);
```

Cierra un controlador de imagen y libera los recursos.

Parámetro	Descripción
identificador	El controlador de imagen.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si hay otra llamada API en curso en la imagen, especialmente si la imagen todavía se está ejecutando. Si se completa la operación, se cierra la imagen y no se puede utilizar el controlador en ninguna llamada API posterior, excepto en otra llamada de `clemrtl_closelImage()`, lo que no tendría efecto alguno.

enumerateParameters

```
int clemrtl_enumerateParameters(
    clemrtl_image_handle controlador,
    clemrtl_parameter_proc procedimiento,
    void* datos);
```


Aplica un procedimiento de devolución de llamada a cada valor y nombre del parámetro de la imagen.

Parámetro	Descripción
identificador	El controlador de imagen.
procedimiento	El procedimiento de la devolución de llamada del parámetro.
datos	Datos especificados por el usuario para la devolución de llamada.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

El procedimiento de devolución de llamada tiene el siguiente tipo:

```
typedef void (*clemrtl_parameter_proc)(
    void* datos,
    const char* nombre,
    const char* valor);
```

Parámetro	Descripción
datos	Datos especificados por el usuario pasados a <code>clemrtl_enumerateParameters()</code> .
nombre	El nombre del parámetro.
valor	El valor del parámetro.

El procedimiento se aplica exactamente una vez a cada parámetro de imagen en un orden arbitrario.

getParameter

```
int clemrtl_getParameter(
    clemrtl_image_handle controlador,
    const char* nombre,
    char* valor,
    int tamaño_valor);
```

Recupera el valor de un parámetro de imagen.

Parámetro	Descripción
identificador	El controlador de imagen.
nombre	El nombre del parámetro.
valor	Un búfer para recibir el valor del parámetro.
tamaño_valor	El tamaño del búfer del valor.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el nombre del parámetro no coincide con el nombre de ningún parámetro de la imagen o si el búfer del valor no es lo suficientemente grande como para recibir el valor del parámetro que incluye un carácter nulo en finalización. Los nombres de los parámetros utilizan el formato `name.attribute` descrito anteriormente (por ejemplo, `file0.name`) y los valores de los parámetros son siempre cadenas, incluso cuando la interpretación es numérica.

setParameter

```
int clemrtl_setParameter(
    clemrtl_image_handle controlador,
    const char* nombre,
    const char* valor);
```

Establece el valor de un parámetro de imagen.

Parámetro	Descripción
identificador	El controlador de imagen.
nombre	El nombre del parámetro.
valor	El valor del parámetro.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el controlador de imagen o si el nombre del parámetro no coincide con el nombre de ningún parámetro en la imagen. Los nombres de los parámetros utilizan el formato `name.attribute` descrito anteriormente (por ejemplo, `file0.name`) y los valores de los parámetros son siempre cadenas, incluso cuando la interpretación es numérica.

getFieldCount

```
int clemrtl_getFieldCount(
    clemrtl_image_handle controlador,
    const char* clave,
    size_t* recuento_campos);
```

Devuelve el número de campos de un origen de entrada o de salida.

Parámetro	Descripción
identificador	El controlador de imagen.
clave	El nombre de la entrada o salida que se va a comprobar, tal y como se ha utilizado en el archivo de parámetros. La clave puede hacer referencia o a una base de datos.
recuento_campos	Recibe el número de campos.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

getFieldTypes

```
int clemrtl_getFieldTypes(
    clemrtl_image_handle controlador,
    const char* clave,
    size_t recuento_campos,
    int* tipo_campos);
```

Devuelve los tipos de campos de un origen de entrada o de salida.

Parámetro	Descripción
identificador	El controlador de imagen.
clave	El nombre de la entrada o salida que se va a comprobar, tal y como se ha utilizado en el archivo de parámetros. La clave puede hacer referencia o a una base de datos.
recuento_campos	El número de campos que se van a examinar. Debe ser igual o inferior a la longitud de la matriz tipo_datos.
tipo_datos	Una matriz de una longitud de al menos recuento_campos que recibe los tipos de campos. Los tipos del primer campo recuento_campos en la entrada o salida se copian en la matriz. Si recuento_campos es mayor que el número real de campos, los elementos adicionales de la matriz se dejan sin definir. Los valores de los tipos de campo se incluyen en la tabla de los tipos de datos siguiente.

Tipos de datos

tipo_datos debe ser uno de los siguientes elementos:

Tipo	Interpretación	Declarador 'C' normal
STRING	Cadena de caracteres UTF con 8 caracteres nulos en finalización.	const char*
INTEGER	Entero con signo de 32 bits.	int
LONG	Entero con signo de 64 bits.	long long

Tipo	Interpretación	Declarador 'C' normal
REAL	Con punto flotante de 64 bits.	double (doble)
TIME	Entero con signo de 64 bits (segundos desde la medianoche).	long long
DATE	Entero con signo de 64 bits (segundos desde la medianoche de 01/01/1970).	long long
TIMESTAMP	Entero con signo de 64 bits (segundos desde la medianoche de 01/01/1970).	long long

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

setAlternativeInput

```
int clemrtl_setAlternativeInput(
    clemrtl_image_handle controlador,
    const char* clave,
    size_t recuento_campos,
    int tipo_campos,
    void** (*iterator)(void* arg),
    void* arg);
```

Sustituye un origen de entrada de archivo con un origen de entrada alternativo.

Parámetro	Descripción
identificador	El controlador de imagen.
clave	El nombre del origen de entrada original que se utiliza en el archivo de parámetros. El origen de entrada debe ser un archivo en el que la clave siempre tenga la forma de "fileN" en algún entero $N \geq 0$.
recuento_campos	El número de campos en la entrada. El valor debe coincidir exactamente con el número de campos de la entrada del archivo original o la llamada no será válida.
tipo_datos	Una matriz de tipos de una longitud de al menos <code>recuento_campos</code> . El valor <code>tipo_datos[i]</code> es el tipo del campo i y debe ser uno de los valores de la tabla de tipos de datos siguientes. Los tipos deben ser compatibles con los de la entrada del archivo original o la llamada no será válida.

Parámetro	Descripción
iterador	Una función que produce los datos de entrada alternativos. La función se aplica a su argumento de la siguiente forma: void** row = iterator(arg); La función se llama durante su ejecución (en una llamada de <code>clemrtl_execute</code>) y se llama una vez para cada registro de entrada. Un valor NULL indica el final de la entrada, en cuyo caso, la función no se vuelve a llamar y la ejecución finalizará. En caso contrario, el resultado es una matriz de datos de una longitud de al menos <code>recuento_campos</code> donde <code>row[i]</code> proporciona el valor del campo <code>i</code> . Un valor puede ser NULL, de lo contrario debe ser un puntero a un dato cuyo tipo está determinado por el <code>tipo_campos[i]</code> correspondiente. Los punteros deben seguir siendo válidos hasta la siguiente llamada del iterador o hasta el final de la ejecución si ésta termina de forma prematura.
arg	Un argumento opaco que se pasa al iterador en cada llamada.

Tipos de datos

`tipo_campos` debe ser uno de una lista de tipos restringida. [Si desea obtener más información, consulte el tema `getFieldTypes` el p. 29.](#)

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación falla si los tipos de campo no son compatibles con los de la entrada del archivo original o si el recuento de campos no coincide exactamente con el número de campos de la entrada del archivo original.

setAlternativeOutput

```
int clemrtl_setAlternativeOutput(
    clemrtl_image_handle controlador,
    const char* clave,
    size_t recuento_campos,
    int tipo_campos,
    void (*iterator)(void* arg, void** row),
    void* arg);
```

Sustituye un objetivo de salida de archivo con un objetivo de salida alternativo.

Parámetro	Descripción
identificador	El controlador de imagen.
clave	El nombre del objetivo de salida original que se utiliza en el archivo de parámetros. El objetivo de salida debe ser un archivo en el que la clave siempre tenga la forma de “fileN” en algún entero $N \geq 0$.
recuento_campos	El número de campos en la salida. El valor debe coincidir exactamente con el número de campos de la salida del archivo original o la llamada no será válida.
tipo_datos	Una matriz de tipos de una longitud de al menos <code>recuento_campos</code> . El valor <code>tipo_datos[i]</code> es el tipo del campo <code>i</code> y debe ser uno de los valores de la tabla de tipos de datos siguientes. Los tipos deben ser compatibles con los de la salida del archivo original o la llamada no será válida.
iterador	Una función que consume la salida de la imagen. La función se aplica a su argumento y a una fila de datos de la siguiente forma: <code>iterador(arg, row);</code> La función se llama durante su ejecución (en una llamada de <code>clemrtl_execute</code>) y se llama una vez para cada fila de resultados que produce la imagen. Un valor de fila NULL indica el final de la salida después de la cual la función no se vuelve a llamar; una aplicación no se debe basar en esta llamada final y debe vaciar y cerrar cualquier recurso externo, etc., cuando la ejecución ha terminado. En caso contrario, la fila es una matriz de datos de una longitud de al menos <code>recuento_campos</code> donde <code>row[i]</code> proporciona el valor del resultado <code>i</code> . Un valor puede ser NULL, de lo contrario debe ser un puntero a un dato cuyo tipo está determinado por el <code>tipo_datos[i]</code> correspondiente. La función debe copiar cualquier valor de datos necesario porque es posible que no se pueda conservar la memoria una vez se ha devuelto la llamada.
arg	Un argumento opaco que se pasa al iterador en cada llamada.

Tipos de datos

`tipo_campos` debe ser uno de una lista de tipos restringida. [Si desea obtener más información, consulte el tema `getFieldTypes` el p. 29.](#)

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_FAIL	Error sin más detalles disponibles.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación falla si los tipos de campo no son compatibles con los de la salida del archivo original o si el recuento de campos no coincide exactamente con el número de campos de la salida del archivo original.

execute

```
int clemrtl_execute(clemrtl_image_handle controlador);
```

Ejecuta una imagen.

Parámetro	Descripción
identificador	El controlador de imagen.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

Si no se ha preparado la imagen, se prepara primero utilizando los valores del parámetro actual. La operación da error si el controlador de imagen no es válido o si se produce un error durante la preparación o la ejecución. La llamada no se devuelve hasta que la ejecución se completa.

prepare

```
int clemrtl_prepare(clemrtl_image_handle controlador);
```

Prepara una imagen para que se ejecute.

Parámetro	Descripción
identificador	El controlador de imagen.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación da error si el controlador de imagen no es válido o si se produce un error durante la preparación.

Una imagen se debe preparar para poder ejecutarla. Al preparar una imagen se fijan los valores del parámetro en la imagen. `clemrtl_execute()` prepara una imagen automáticamente en cada llamada si aún no se ha preparado; resulta útil utilizar `clemrtl_prepare()` para preparar una imagen si se va a ejecutar la imagen varias veces con los mismos valores del parámetro, y permite mejorar el rendimiento considerablemente. Una vez que se ha preparado una imagen, se ignoran los cambios posteriores de los valores del parámetro; vuelva a utilizar `clemrtl_prepare()` para actualizar la imagen con los valores nuevos del parámetro.

interrupt

```
int clemrtl_interrupt(clemrtl_image_handle controlador);
```

Finaliza la ejecución en curso en una imagen.

Parámetro	Descripción
identificador	El controlador de imagen.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

La operación no tiene efecto alguno si el controlador de imagen no es válido o si la imagen no se está ejecutando.

Es seguro llamar a esta función simultáneamente con otra llamada API en el mismo controlador de imagen.

getErrorDetail

```
int clemrtl_getErrorDetail(
    clemrtl_image_handle controlador,
    char* gravedad,
    int* código,
    char* texto,
    int tamaño_texto);
```

Recupera información detallada sobre el último error que se produjo en una imagen.

Parámetro	Descripción
identificador	El controlador de imagen.
gravedad	Recibe el código de gravedad como un carácter único: I—Información W—Advertencia E—Error X—Error del sistema
código	Recibe el número de error.
texto	Un búfer para recibir el texto del mensaje.
tamaño_texto	Tamaño del búfer del texto.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

Los resultados serán inestables si el controlador de imagen no es válido. El texto del mensaje se divide si es necesario para ajustarlo en el búfer del texto.

Si una llamada API da el error CLEMRTL_ERROR y éste no está relacionado con ningún controlador de imagen en particular, al pasar el valor 0 como el controlador de imagen se recuperarán los detalles del último error no específico de imagen, pero es inestable en un entorno de subprocesamiento múltiple.

setReportHandler

```
int clemrtl_setReportHandler(
    clemrtl_image_handle controlador,
    clemrtl_report_proc procedimiento,
    void* datos);
```

Instala un procedimiento de devolución de llamada de informe para una imagen.

Parámetro	Descripción
identificador	El controlador de imagen.
procedimiento	El procedimiento de devolución de llamada de informe.
datos	Datos especificados por el usuario para la devolución de llamada.

Devuelve uno de los siguientes códigos de estado:

Resultado	Descripción
CLEMRTL_OK	Completado.
CLEMRTL_ERROR	Error con más detalles disponibles.

El procedimiento de devolución de llamada tiene el siguiente tipo:

```
typedef void (*clemrtl_report_proc)(
    void* datos,
    char gravedad,
    int código,
    const char* texto);
```

Parámetro	Descripción
datos	Datos especificados por el usuario pasados a clemrtl_setReportHandler().
gravedad	El código de gravedad como un carácter único: I—Información W—Advertencia E—Error X—Error del sistema
código	El número del mensaje.
texto	El texto del mensaje.

Este procedimiento se aplica a cada mensaje según llega. Al establecer el procedimiento en NULL se elimina cualquier controlador instalado en la imagen.

Avisos

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.

El párrafo siguiente no se aplica en el Reino Unido ni en cualquier otro país en los que dichas provisiones sean contrarias a la legislación local: SPSS INC., AN IBM COMPANY, PROPORCIONA ESTA PUBLICACIÓN “TAL CUAL” SIN GARANTÍAS DE NINGÚN TIPO, YA SEA EXPRESAS O IMPLÍCITAS, INCLUYENDO, SIN LIMITAR LA GENERALIDAD DE LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten el descargo de responsabilidad de garantías expresas o implícitas en determinadas transacciones, por lo que esta declaración no será aplicable.

Esta información puede incluir imprecisiones técnicas o errores tipográficos. La información que se contiene se puede modificar periódicamente; estos cambios se incorporarán en las nuevas ediciones de la publicación. SPSS Inc. puede realizar mejoras y/o cambios en el producto(s) y/o el programa(s) descrito en esta publicación en cualquier momento sin notificación.

Las referencias a esta información en sitios web ajenos a SPSS y a IBM se proporcionan únicamente por motivos de comodidad y no servirán de ninguna forma como aprobación de esos sitios web. Los materiales de esos sitios web no forman parte de los materiales de este producto de SPSS Inc. y el uso de esos sitios web se realiza bajo su responsabilidad.

Al enviar información a IBM o SPSS, el usuario concede a IBM y a SPSS el derecho no exclusivo de utilizar o distribuir la información de la forma que estime adecuada sin incurrir en obligaciones con el usuario.

La información relacionada con productos ajenos a productos SPSS se ha obtenido de los proveedores de esos productos, de sus anuncios publicados u otros orígenes disponibles de forma pública. SPSS no ha comprobado esos productos y no puede confirmar la precisión del rendimiento, compatibilidad o cualquier otras reclamaciones relacionadas con productos ajenos a SPSS. Las cuestiones sobre las responsabilidades de productos ajenos a SPSS se deben dirigir a los proveedores de esos productos.

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.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales habituales. Para ilustrarlas de la forma más completa posible, los ejemplos incluyen los nombres de personas, empresas, marcas y productos. Todos estos nombres son inventados y cualquier similitud con los nombres y direcciones de una empresa real es una coincidencia.

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

Marcas comerciales

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas comerciales de IBM Corporation, registradas en múltiples jurisdicciones en todo el mundo. Existe una lista actualizada de las marcas comerciales de IBM disponible en Internet en <http://www.ibm.com/legal/copytrade.shtml>.

SPSS es una marca comercial de SPSS Inc., an IBM Company, registradas en múltiples jurisdicciones en todo el mundo.

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript son marcas comerciales registradas o marcas comerciales de Adobe Systems Incorporated en los Estados Unidos y/o en otros países.

IT Infrastructure Library es una marca comercial registrada de la Agencia central de telecomunicaciones y computación central que ahora forma parte de la Oficina de comercio gubernamental.

Intel, el logotipo de Intel logo, Intel Inside, el logotipo de Intel, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas comerciales o marcas comerciales registradas de Intel Corporation o de sus filiales en los Estados Unidos y en otros países.

Linux es una marca comercial registrada de Linus Torvalds en los Estados Unidos, en otros países o ambos.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas comerciales de Microsoft Corporation en los Estados Unidos, en otros países o ambos.

ITIL es una marca comercial registrada y una marca comercial comunitaria registrada de la Oficina de Comercio Gubernamental y está registrada en la Oficina de patentes y marcas comerciales de los Estados Unidos.

UNIX es una marca comercial registrada de The Open Group en los Estados Unidos y en otros países.

Cell Broadband Engine es una marca comercial de Sony Computer Entertainment, Inc. en los Estados Unidos, en otros países o ambos y se utiliza con licencia.

Java y todas las marcas comerciales y logotipos basados en Java son marcas comerciales de Sun Microsystems, Inc. en los Estados Unidos, en otros países o ambos.

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.

Otros nombres de productos y servicios pueden ser marcas comerciales de IBM, SPSS u otras empresas.

- apertura
 - rutas, 25
- API
 - códigos de error, 21
 - códigos de estado, 21
 - conceptos básicos, 12, 21
 - entrada alternativa, 30
 - opciones de ejecución, 24
 - recuento de campos, 28
 - referencia, 21
 - salida alternativa, 31
 - setReportHandler, 35
 - tipos de campos, 29
- archivo *.par*, 7
- archivo *pim*, 7
- archivos de registro, 25
- avisos legales, 36

- biblioteca de programación de tiempo de ejecución (CLEMRTL), 12

- cierre
 - rutas, 26
- closeImage
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 26
- códigos de error, 21
- códigos de estado, 21
- configuración regional
 - definir mediante la API, 22

- desinstalación
 - IBM SPSS Modeler Solution Publisher Runtime, 20
- directorio temporal
 - IBM SPSS Modeler Solution Publisher Runtime, 19
- distribución, 6
- documentación, 2

- ejecución de rutas
 - a través de IBM SPSS Modeler Solution Publisher, 10–11
- ejemplos
 - conceptos básicos, 4
 - Manual de aplicaciones, 2
- ejemplos de aplicaciones, 2
- enumerateParameters
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 26
- execute
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 33

- funciones 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
 - inicialización, 22
 - initializing, 21
 - 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

- getErrorDetail
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 34
- getFieldCount
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 28
- getFieldTypes
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 29
- getOption
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 24
- getParameter
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 27

- IBM SPSS Modeler, 1
 - documentación, 2
- IBM SPSS Modeler Solution Publisher, 6–7
 - conceptos básicos, 6
 - incrustación en aplicaciones, 12
- IBM SPSS Modeler Solution Publisher Runtime
 - desinstalación, 20
 - directorio temporal, 19
 - inicio, 19
 - instalación, 15
 - requisitos del sistema, 15–16
- IBM SPSS Text Analytics, 2
- initialise
 - marcas, 22
 - referencia de funciones API, 22
- initialise_ext
 - marcas, 22
 - referencia de funciones API, 22
- initialize
 - conceptos básicos del proceso de API, 21

- interrupt
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 33

- marcas comerciales, 37

- nodos de resultados, 6
- notificación de errores
 - referencia de funciones API, 25, 34–35

- openImage
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 25

- parameters
 - parámetros de ejecución de ruta, 11
- parámetros
 - archivos de parámetros de API, 25–26
 - parámetros de imagen de API, 26–28
 - procedimientos de devolución de llamadas API, 26
- prepare
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 33
- publicación de rutas
 - IBM SPSS Modeler Solution Publisher, 6–7

- Runtime
 - desinstalación, 20
 - directorio temporal, 19
 - inicio, 19
 - instalación de Windows, 16
 - instalación en UNIX, 16
 - opciones, 10

- setAlternativeInput
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 30
- setAlternativeOutput
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 31
- setLogFile
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 25
- setOption
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 24
- setParameter
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 28
- setReportHandler
 - conceptos básicos del proceso de API, 21
 - referencia de funciones API, 35
- SPSS Modeler Server, 1

- UNIX
 - instalación de IBM SPSS Modeler Solution Publisher Runtime, 16

- Windows
 - instalación de IBM SPSS Modeler Solution Publisher Runtime, 15

- zona horaria
 - definir mediante la API, 22