

*IBM SPSS Modeler 18.1
Solution Publisher*

IBM

Примечание

Прежде чем использовать эту информацию и продукт, описанный в ней, прочтите сведения в разделе “Уведомления” на стр. 37.

Информация о продукте

Это издание применимо к версии 18, выпуск 0, модификация 0 IBM SPSS Modeler и ко всем последующим версиям и модификациям до тех пор, пока в новых изданиях не будет указано иное.

Содержание

Предисловие	v	Глава 4. Установка модуль времени выполнения IBM SPSS Modeler Solution Publisher.	15
Глава 1. О программе IBM SPSS Modeler.	1	Установка модуля времени выполнения	15
Продукты IBM SPSS Modeler	1	Установка в Windows	15
IBM SPSS Modeler.	1	Установка в UNIX	15
IBM SPSS Modeler Server	1	Устранение неполадок при установке	16
IBM SPSS Modeler Administration Console	2	Запуск модуля времени выполнения	17
IBM SPSS Modeler Batch.	2	Изменение каталога temp	17
IBM SPSS Modeler Solution Publisher	2	Деинсталляция модуля времени выполнения IBM SPSS Modeler Solution Publisher	18
Адаптеры IBM SPSS Modeler Server для IBM SPSS Collaboration and Deployment Services.	2	Процедура деинсталляции в Windows	18
Выпуски IBM SPSS Modeler.	2	Процедура деинсталляции в UNIX	18
Документация IBM SPSS Modeler.	3	Глава 5. API встроенного предсказательного моделирования IBM SPSS	19
Документация SPSS Modeler Professional	3	Глава 6. Справочник по API библиотеки времени выполнения IBM SPSS Modeler Solution Publisher	21
Документация SPSS Modeler Premium	4	Введение	21
Примеры прикладных программ	4	Обзор процессов API	21
Папка demos	4	Функции API	22
Отслеживание лицензий	4	Уведомления.	37
Глава 2. IBM SPSS Modeler Solution Publisher	7	Товарные знаки	38
Как работает IBM SPSS Modeler Solution Publisher	7	Правила и условия для документации продукта.	39
Публикация потоков	8	Индекс	41
Глава 3. Выполнение опубликованных потоков	11		
Настройка выполнения потока с помощью файла параметров	12		
Встраивание IBM SPSS Modeler Solution Publisher в прикладные программы	13		
Пример прикладной программы IBM SPSS Modeler Solution Publisher	14		

Предисловие

IBM® SPSS Modeler - это IBM Corp. инструментальная среда масштаба предприятия для анализа данных. SPSS Modeler помогает организациям улучшить взаимосвязи с клиентами и отдельными лицами, обеспечивая глубокое понимание данных. Организации используют приобретенные с помощью SPSS Modeler глубокие знания для сохранения выгодных заказчиков, обнаружения возможностей дополнительных покупок, привлечения новых клиентов, обнаружения ошибок, сокращения рисков и улучшений в обеспечении государственных служб.

Наглядный интерфейс SPSS Modeler дает пользователям возможность применить свой конкретный опыт в бизнесе, что способствует разработке более мощных предсказывающих моделей и сокращает время принятия решения. SPSS Modeler предлагает много способов моделирования, таких как алгоритмы предсказания, классификации, сегментации и ассоциативного обнаружения. Когда моделей IBM SPSS Modeler Solution Publisher поддерживает их распространение на уровне организации для принимающих решение сотрудников или для применения к базе данных.

О бизнес аналитике IBM

Программное обеспечение IBM для бизнес аналитики предоставляет полную, последовательную и точную информацию, которая повышает эффективность ведения бизнеса. Полный набор программного обеспечения для business intelligence, прогностической аналитики, управления финансовой эффективностью и стратегией и аналитических приложений позволяет ясно видеть текущую ситуацию, а также делать прогнозы, позволяющие предпринимать практические действия. В сочетании с решениями для конкретных отраслей, проверенной практикой и услугами бизнес аналитика IBM позволяет организациям любых размеров достигать наивысшей производительности, уверенно автоматизировать процессы принятия решений и добиться лучших результатов.

Как составная часть этого набора, программное обеспечение IBM SPSS Predictive Analytics помогает организациям предсказывать будущие события и предпринимать практические действия непосредственно на основе этих предсказаний. Коммерческие, правительственные и академические организации всего мира, полагаются на технологию IBM SPSS, обеспечивающую конкурентное преимущество в привлечении, удержании и повышении отдачи от клиентов. Включая программное обеспечение IBM SPSS в свои ежедневные операции, организации могут прогнозировать будущие события, направлять и автоматизировать решения для соответствия бизнес-целям и достигать ощутимых конкурентных преимуществ. Чтобы получить дальнейшую информацию или связаться с представителем, зайдите на <http://www.ibm.com/spss>.

Техническая поддержка

Техническая поддержка предоставляется клиентам, оплачивающим обновительные взносы. Пользователи могут обращаться в службу технической поддержки, если у них возникают какие-либо проблемы с использованием или установкой программного обеспечения IBM Corp.. За технической поддержкой обращайтесь на сайт IBM Corp.: <http://www.ibm.com/support>. При обращении за поддержкой будьте готовы назвать себя и организацию, в которой вы работаете.

Глава 1. О программе IBM SPSS Modeler

IBM SPSS Modeler - это комплект инструментов исследования данных, при помощи которого можно быстро разрабатывать прогнозные модели, использующие деловые знания и опыт, и внедрять их в деловые операции для усовершенствования процесса принятия решений. Разработанный на основе модели промышленного стандарта CRISP-DM, IBM SPSS Modeler поддерживает весь процесс исследования данных, от обработки исходных данных до получения лучших деловых результатов.

IBM SPSS Modeler предлагает ряд методов моделирования, взятых из таких областей, как обучение машин, искусственный интеллект и статистика. При помощи методов, доступных на палитре Моделирование, можно извлечь новую информацию из данных и разработать прогнозные модели. У каждого из методов есть свои сильные стороны и типы задач, для решения которых он лучше всего подходит.

SPSS Modeler можно приобрести как отдельный продукт или использовать как клиент в сочетании с SPSS Modeler Server. Кроме того, доступен ряд дополнительных возможностей, сводка которых дается в следующих разделах. Дополнительную информацию смотрите по ссылке <https://www.ibm.com/analytics/us/en/technology/spss/>.

Продукты IBM SPSS Modeler

В семейство продуктов IBM SPSS Modeler и связанные с этим семейством программы входят следующие продукты:

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (включено в IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- Адаптеры IBM SPSS Modeler Server для IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler - это полнофункциональная версия продукта, устанавливаемая и запускаемая на персональном компьютере. SPSS Modeler можно запустить в локальном режиме, как автономный продукт, или в распределенном режиме вместе с IBM SPSS Modeler Server, чтобы повысить производительность на больших наборах данных.

Используя SPSS Modeler, можно быстро и интуитивно строить точные прогнозные модели, не прибегая к программированию. Используя уникальный визуальный интерфейс, можно легко визуализировать процесс анализа данных. В продукт встроены расширенные функции аналитики, при поддержке которых можно обнаруживать в данных скрытые структуры и тенденции. Можно моделировать результаты и выяснять, какие факторы на них влияют, чтобы полностью использовать деловые возможности и ограничивать риски.

SPSS Modeler доступен в двух версиях: SPSS Modeler Professional и SPSS Modeler Premium. Дополнительную информацию смотрите в разделе “Выпуски IBM SPSS Modeler” на стр. 2.

IBM SPSS Modeler Server

SPSS Modeler пользуется архитектурой клиент - сервер, чтобы распределять требования ресурсоемких операций по мощным серверным программам, что повышает производительность для больших наборов данных.

SPSS Modeler Server - это отдельно лицензируемый продукт, который непрерывно работает в режиме распределенного анализа на хосте сервера совместно с одной или несколькими установками IBM SPSS Modeler. При этом SPSS Modeler Server обеспечивает высокую производительность для больших наборов данных, поскольку ресурсоемкие операции можно выполнять на сервере без скачивания данных на компьютер клиента. Кроме того, IBM SPSS Modeler Server обеспечивает поддержку для возможностей оптимизации SQL и моделирования в базе данных, что дает дополнительный выигрыш в производительности и автоматизации.

IBM SPSS Modeler Administration Console

Modeler Administration Console - это графический пользовательский интерфейс для управления многочисленными опциями конфигурации SPSS Modeler Server; их можно также конфигурировать и посредством файла опций. Консоль, входящая в состав IBM SPSS Deployment Manager, может использоваться для отслеживания и конфигурирования установок SPSS Modeler Server; она доступна без дополнительной оплаты для действующих заказчиков SPSS Modeler Server. Эту прикладную программу можно установить только на компьютерах Windows; однако она может управлять сервером на любой поддерживаемой платформе.

IBM SPSS Modeler Batch

Хотя обычно исследование данных - интерактивный процесс, можно также запустить SPSS Modeler из командной строки, не открывая графический интерфейс. Например, у вас могут быть продолжительные или повторяющиеся задачи, которые желательно выполнить без участия пользователя. SPSS Modeler Batch - это особая версия продукта, предоставляющая поддержку всех аналитических возможностей SPSS Modeler без вызова обычного пользовательского интерфейса. SPSS Modeler Server необходим для использования SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher - это инструмент, при помощи которого можно создать пакетную версию потока SPSS Modeler; такую версию можно запускать внешним механизмом времени выполнения или встроить во внешнюю прикладную программу. Этим способом можно публиковать и внедрять полные потоки SPSS Modeler для использования в средах, где SPSS Modeler не установлен. SPSS Modeler Solution Publisher распространяется в составе службы IBM SPSS Collaboration and Deployment Services - Scoring, для которой требуется отдельная лицензия. С этой лицензией вы получаете SPSS Modeler Solution Publisher Runtime, при помощи которого можете запускать опубликованные потоки.

Дополнительную информацию о SPSS Modeler Solution Publisher смотрите в документации IBM SPSS Collaboration and Deployment Services. Центр знаний IBM SPSS Collaboration and Deployment Services содержит разделы "IBM SPSS Modeler Solution Publisher" и "IBM SPSS Analytics Toolkit".

Адаптеры IBM SPSS Modeler Server для IBM SPSS Collaboration and Deployment Services

Для IBM SPSS Collaboration and Deployment Services доступен ряд адаптеров, при посредстве которых SPSS Modeler и SPSS Modeler Server могут взаимодействовать с репозиторием IBM SPSS Collaboration and Deployment Services. При этом поток SPSS Modeler, внедренный в репозиторий, доступен для совместного использования несколькими пользователями или для обращения из прикладной программы IBM SPSS Modeler Advantage тонкого клиента. Адаптер устанавливается в той системе, в которой находится репозиторий.

Выпуски IBM SPSS Modeler

SPSS Modeler доступен в следующих выпусках.

SPSS Modeler Professional

SPSS Modeler Professional содержит все инструменты, необходимые для работы с большинством типов структурированных данных, таких как трассировка поведения и взаимодействия в системах CRM, демографии, поведения покупателей и данных о продажах.

SPSS Modeler Premium

SPSS Modeler Premium - это отдельно лицензируемый продукт, расширяющий SPSS Modeler Professional для работы со специализированными данными и с неструктурированными текстовыми данными. SPSS Modeler Premium включает в себя IBM SPSS Modeler Text Analytics:

IBM SPSS Modeler Text Analytics использует новейшие лингвистические технологии и обработку естественного языка (NLP) для быстрой обработки самых разнообразных неструктурированных текстовых данных, для извлечения и организации ключевых понятий и группирования этих понятий в категории. Извлеченные понятия и категории можно сочетать с существующими структурированными данными, такими как демографические, и применять к моделированию при помощи полного комплекта инструментов исследования данных IBM SPSS Modeler для получения более качественных и специализированных решений.

Документация IBM SPSS Modeler

К документации можно обратиться из меню Справка в SPSS Modeler. При этом открывается Центр знаний SPSS Modeler, общедоступный вне данного продукта.

Полная документация для каждого продукта (в том числе инструкции по установке) доступна также в формате PDF в нескольких сжатых папках как часть скачиваемого образа продукта. Кроме этого, можно скачать документы PDF с веб-сайта <http://www.ibm.com/support/docview.wss?uid=swg27046871>.

Документация SPSS Modeler Professional

В комплект документации SPSS Modeler Professional (включая указания по установке) входят:

- **IBM SPSS Modeler Руководство пользователя.** Общее введение в использование SPSS Modeler, в том числе о создании потоков данных, обработке пропущенных значений, построению выражений CLEM, работе с проектами и отчетами и составлению пакетов потоков для внедрения в IBM SPSS Collaboration and Deployment Services или IBM SPSS Modeler Advantage.
- **Узлы источников, обработки и вывода IBM SPSS Modeler.** Описания всех узлов, служащих для чтения, обработки и вывода данных в различных форматах. По существу это все узлы, кроме узлов моделирования.
- **Узлы моделирования IBM SPSS Modeler.** Описания всех узлов, служащих для создания моделей исследования данных. IBM SPSS Modeler предлагает ряд методов моделирования, взятых из таких областей, как обучение машин, искусственный интеллект и статистика.
- **Руководство по прикладным программам IBM SPSS Modeler.** Примеры в этом руководстве служат кратким специализированным введением к тем или иным методам и технологиям моделирования. Это руководство доступно также в электронном виде в меню Справка. Дополнительную информацию смотрите в разделе “Примеры прикладных программ” на стр. 4.
- **Сценарии и автоматизация Python IBM SPSS Modeler.** Информация об автоматизации системы путем создания сценариев Python, включая сценарии свойств, которые могут использоваться для работы с узлами и потоками.
- **Руководство по внедрению IBM SPSS Modeler .** Информация о выполнении IBM SPSS Modeler потоков как шагов обработки заданий под управлением IBM SPSS Deployment Manager.
- **Руководство разработчика IBM SPSS Modeler CLEF .** CLEF предоставляет возможности интеграции с программами других производителей, таких как подпрограммы обработки данных или алгоритмы моделирования, как с узлами в IBM SPSS Modeler.

- **Руководство по исследованию данных в базе данных IBM SPSS Modeler.** Информация о том, как использовать мощности вашей базы данных для повышения производительности и расширения диапазона возможностей анализа с привлечением алгоритмов от сторонних производителей.
- **Руководство администратора и руководство по производительности IBM SPSS Modeler Server .** Информация о том, как сконфигурировать и администрировать IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager Руководство пользователя.** Информация об использовании пользовательского интерфейса административной консоли включено в прикладную программу Deployment Manager для мониторинга и конфигурирования сервера IBM SPSS Modeler.
- **Руководство по CRISP-DM IBM SPSS Modeler.** Пошаговое руководство к использованию методологии CRISP-DM для исследования данных SPSS Modeler.
- **IBM SPSS Modeler Batch Руководство пользователя.** Полное руководство по использованию IBM SPSS Modeler в пакетном режиме, включая подробности выполнения в пакетном режиме и аргументы командной строки. Это руководство доступно только в формате PDF.

Документация SPSS Modeler Premium

В комплект документации SPSS Modeler Premium (включая указания по установке) входят:

- **SPSS Modeler Text Analytics Руководство пользователя.** Информация об использовании аналитики текстов совместно с SPSS Modeler, в том числе по узлам исследования текстов, интерактивной инструментальной среде, шаблонам и другим ресурсам.

Примеры прикладных программ

Инструменты исследования данных в SPSS Modeler помогают разрешить широкий спектр деловых и организационных проблем, а примеры прикладных программ предоставляют краткие, целевые введения в конкретные методы и способы моделирования. Используемые здесь наборы данных намного меньше огромных складов данных, которыми управляют некоторые исследователи данных, но применяемые понятия и методы должны масштабироваться до реальных прикладных программ.

Чтобы обратиться к примерам, выберите **Примеры прикладных программ** в меню Справка в SPSS Modeler.

Файлы данных и потоки примеров устанавливаются в папке Demos в каталоге установки продукта. Дополнительную информацию смотрите в разделе “Папка demos”.

Примеры моделирования баз данных. Смотрите эти примеры в руководстве *IBM SPSS Modeler In-Database Mining Guide*.

Примеры сценариев. Смотрите эти примеры в руководстве *IBM SPSS Modeler Scripting and Automation Guide*.

Папка demos

Файлы данных и примеры потоков, используемые с примерами прикладных программ, устанавливаются в папке Demos в каталоге установки продукта (например: C:\Program Files\IBM\SPSS\Modeler\<версия>\Demos). К этой папке можно также обратиться из группы программ IBM SPSS Modeler в меню Пуск Windows или, щелкнув по Demos в списке недавно использовавшихся каталогов в диалоговом окне **Файл > Открыть поток**.

Отслеживание лицензий

При работе с SPSS Modeler использование лицензий отслеживается и записывается в журнал через регулярные интервалы времени. В журнал записываются показатели лицензирования *AUTHORIZED_USER* и *CONCURRENT_USER*; тип записываемого в журнал показателя зависит от типа лицензии, которая у вас есть для SPSS Modeler.

Генерируемые файлы журналов могут обрабатываться инструментом IBM License Metric Tool, из которого вы можете сгенерировать отчеты об использовании лицензий.

Файлы журналов лицензирования создаются в том же каталоге, куда записываются и файлы журналов клиента SPSS Modeler (по умолчанию %ALLUSERSPROFILE%/9IBM/SPSS/Modeler/<версия>/log).

Глава 2. IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher - мощный инструмент интеграции ваших результатов исследования данных в бизнес-процесс для решения реальных проблем. С использованием IBM SPSS Modeler Solution Publisher можно создать "пакетную" версию потока, который можно выполнить при помощи внешнего механизма времени выполнения или встроить во внешнюю программу. Это позволяет внедрить потоки моделирования данных в производственную среду для поддержки каждодневных бизнес-процессов и для предоставления лицам, принимающим решения, знаний, полученных при исследовании данных.

Использование IBM SPSS Modeler Solution Publisher предоставляет больше возможностей, чем простой экспорт модели (в виде PMML), поскольку оно позволяет вам публиковать и внедрять *полные потоки IBM SPSS Modeler*. Это означает, что вы можете выполнять подготовку данных, так же, как и операции с записями и полями, такие как агрегирование данных, выбор записей и построение новых производных полей, перед созданием прогнозов на основе модели. После этого вы можете выполнить дальнейшую обработку результатов модели перед сохранением данных - и все это просто путем выполнения опубликованного потока.

Примечание: Публикация выполняется при помощи вкладки Опубликовать соответствующего узла экспорта. Дополнительную информацию смотрите в разделе “Публикация потоков” на стр. 8. Потоки, сохраненные в выпусках до версии 11.0, использовали отдельный узел Публикатор. Этот узел по-прежнему поддерживается из соображений обратной совместимости, но он более не доступен на палитре узлов.

Лицензирование IBM SPSS Modeler Solution Publisher

Примечание: IBM SPSS Modeler Solution Publisher распространяется в составе продукта IBM SPSS Collaboration and Deployment Services Scoring Service, для которого требуется отдельная лицензия. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Когда вы приобретаете лицензию, поставляется отдельный модуль времени выполнения IBM SPSS Modeler Solution Publisher, который позволяет выполнять опубликованные потоки. Дополнительную информацию смотрите в разделе “Установка модуля времени выполнения” на стр. 15.

Как работает IBM SPSS Modeler Solution Publisher

Внедрение решения при помощи IBM SPSS Modeler Solution Publisher включает две фазы: *публикация* потока и *выполнение* потока.

Публикация. В процессе исследования данных в какой-то момент вы придете к модели, хорошо подходящей для решения ваших бизнес-задач. В этот момент вы готовы принять эту модель и применить ее к вашему бизнес-процессу. При публикации потока его подробное описание записывается на диск (в виде файла *образа* и файла *параметров*). Дополнительную информацию смотрите в разделе “Публикация потоков” на стр. 8.

Выполнение. После публикации потока можно воссоздать процесс, реализованный в этом потоке, путем выполнения опубликованного потока. Это делается либо при помощи автономного модуля времени выполнения IBM SPSS Modeler (*modelerrun.exe*), либо путем разработки прикладной программы, использующей для выполнения потока библиотеку времени выполнения IBM SPSS Modeler. Для выполнения потоков за пределами IBM SPSS Modeler (при помощи модуля времени выполнения или пользовательской прикладной программы) необходимо сначала установить модуль времени выполнения IBM SPSS Modeler Solution Publisher. Дополнительную информацию смотрите в разделе “Установка модуля времени выполнения” на стр. 15.

Публикация потоков

Публикация потоков выполняется непосредственно из IBM SPSS Modeler с использованием любого из стандартных узлов экспорта: узлы Database (База данных), Flat File (Плоский файл), Экспорт Statistics, Extension Export (Экспорт расширений), Data Collection Export (Экспорт собраний данных), Экспорт SAS, Excel и Экспорт XML. Тип узла экспорта определяет формат записи результатов при каждом выполнении опубликованного потока с помощью модуля времени выполнения модуль времени выполнения IBM SPSS Modeler Solution Publisher или внешней программы. Например, если вы хотите записывать результаты в базу данных при каждом выполнении опубликованного потока, используйте узел экспорта Database (База данных).

Опубликовать сценарий

1. Откройте или постройте поток обычным способом и присоедините узел экспорта в конце.
2. На вкладке Опубликовать узла экспорта укажите корневое имя публикуемых файлов, то есть имя, к которому будут присоединяться расширения .rim, .par и .xml).
3. Щелкните по **Опубликовать**, чтобы опубликовать поток, или выберите опцию **Опубликовать поток** для автоматической публикации потока всякий раз, когда выполняется узел.

Опубликованное имя. Укажите корневое имя опубликованного образа и файлы параметров.

- **Файл образа (*.rim)** предоставляет всю информацию, необходимую модулю времени выполнения для выполнения опубликованного потока в точности таким, каким он был в момент экспорта. Если вы уверены, что вам не потребуются изменять какие-либо параметры для этого потока (например, входной источник данных или выходной файл данных), можно внедрять только файл образа.
- **Файл параметров (*.par)** содержит конфигурируемую информацию об источниках данных, выходных файлах и опциях выполнения. Если вы хотите иметь возможность управлять вводом или выводом потока без повторной публикации этого потока, в дополнение к файлу образа вам потребуется файл параметров.
- **Файл метаданных (*.xml)** описывает ввод и вывод образа и их модели данных. Он предназначен для использования прикладными программами, встраивающими библиотеку времени выполнения, которым нужно знать структуру данных ввода и вывода.

Примечание: Этот файл создается только в том случае, если вы выбрали опцию **Опубликовать метаданные**.

Опубликовать параметры. При необходимости можно включить в файл *.par параметры потока. Эти значения параметров потока можно изменять при выполнении образа, или изменяя файл *.par, или через API среды выполнения.

Эта опция включает кнопку **Параметры**. При нажатии этой кнопки выводится диалоговое окно Опубликовать параметры.

Определите параметры, которые вы хотите включить в опубликованный образ, выбрав соответствующую опцию в столбце **Опубликовать**.

Выполнение в потоке. Указывает, нужно ли автоматически публиковать поток при выполнении узла.

- **Экспортировать данные.** Выполняет узел экспорта обычным способом, не публикуя поток. (В основном узел исполняется в IBM SPSS Modeler так же, как это было бы при недоступности IBM SPSS Modeler Solution Publisher.) При выборе этой опции поток не будет публиковаться, пока вы не сделаете этого явно, щелкнув по **Опубликовать** в диалоговом окне узла экспорта. Другой вариант - опубликовать текущий поток при помощи инструмента Опубликовать на панели инструментов или при помощи сценария.
- **Опубликовать поток.** Публикует поток для внедрения при помощи IBM SPSS Modeler Solution Publisher. Включите эту опцию, если вы хотите автоматически публиковать поток каждый раз, когда он выполняется.

Примечание:

- Если вы планируете выполнять опубликованный поток с новыми или обновленными данными, важно отметить, что порядок полей во входном файле должен совпадать с порядком полей во входном файле узла источника, определенном в опубликованном потоке.
- При публикации во внешние прикладные программы вам может потребоваться отфильтровать лишние поля или переименовать поля в соответствии с требованиями ввода этих программ. И то, и другое можно обеспечить с помощью узла Фильтр перед узлом экспорта.

Глава 3. Выполнение опубликованных потоков

Основной эффект выполнения опубликованного потока всегда состоит в генерировании набора данных. Эти данные могут сохраняться в файл на диске или записываться в базу данных. Опубликованные потоки не могут генерировать графики, таблицы, генерируемые модели или другие типы вывода, не являющиеся данными.

Потоки, опубликованные при помощи IBM SPSS Modeler Solution Publisher, можно выполнять с помощью программы модуля времени выполнения IBM SPSS Modeler Solution Publisher. Программа модуля времени выполнения, *modelerrun.exe*, запускается из командной строки, с опциями, указывающими файл образа и (необязательно) файл параметров для выполнения. Синтаксис команды модуля времени выполнения:

```
modelerrun -nobanner -p <файл-параметров> -o <опции> <файл-образа>
```

Описание параметров команды дано в следующей таблице.

Таблица 1. Параметры команды модуля времени выполнения

Параметр команды	Описание
-nobanner	подавляет текстовое сообщение при запуске модуля времени выполнения (необязательно).
<файл-параметров>	Имя файла параметров (необязательно).
<опции>	список пар вида опция=значение через запятую, указывающих опции выполнения для модуля времени выполнения. Опции подробно описаны ниже (необязательно). Примечание: После запятой не должно быть пробельных символов.
<файл-образа>	имя файла образа для выполнения (обязательно).

Доступные опции выполнения описаны в следующей таблице.

Таблица 2. Допустимые опции модуля времени выполнения.

Имя	Значение по умолчанию	Описание
max_file_size	-1	Максимальный размер сохраненных файлов. Значение -1 означает отсутствие предела.
max_sql_string_length	2048	Максимальная длина строки с SQL, импортированной из базы данных. Более длинные строковые значения усекаются справа без предупреждения. Допустимый диапазон - от 1 до 65535 символов.
memory_usage	100	Коэффициент для динамического выделения памяти. Увеличьте или уменьшите это значение, чтобы управлять общим использованием памяти на сервере
temp_directory	"	Задаёт каталог, используемый для временных файлов.
request_passwords	Y	Определяет, выводится ли предложение ввести пароль к базе данных при выполнении опубликованного потока, требующего доступа к базе данных (Y - да, N - нет).

Запуск локального потока Apache Spark в IBM SPSS Modeler Solution Publisher

При запуске локального потока Spark в IBM SPSS Modeler Solution Publisher с командой `modelerrun` необходимо указать следующее (где `C:/Anaconda2/python.exe` - это путь исполняемого файла Python):
`-o eas_pyspark_python_path=C:/Anaconda2/python.exe`

Настройка выполнения потока с помощью файла параметров

Когда поток опубликован, файл образа содержит информацию об источнике данных и о назначении данных вывода, выбранных в первоначальном потоке. Однако часто бывает удобно запустить поток для данных из другого источника данных или перенаправить данные вывода в другое назначение. Это можно сделать путем редактирования файла параметров потока.

Это особенно важно, если вы будете выполнять опубликованный поток на другом компьютере, отличающемся от компьютера, на котором поток был создан. Необходимо будет обновить положения файлов ввода и вывода в соответствии со структурой компьютера назначения. Обратите внимание на то, что опубликованные потоки не поддерживают относительные пути, что делает особенно важной проверку положений данных в вашем файле параметров. (Если при публикации потока вы укажете относительные пути в опциях узла экспорта, IBM SPSS Modeler перед созданием файлов образа и параметров для опубликованного потока преобразует их в абсолютные пути.)

Кроме того, в файле параметров можно задать опции выполнения и изменить значения параметров потока.

Значения параметров задаются в файле параметров в виде пар `<имя.атрибут>=<значение>`. Для некоторых параметров имя параметра содержит цифру ID для различения узлов одного типа в пределах потока. Например, в потоке, где и ввод, и вывод указывают на соединение с базой данных, параметры, описывающие входное соединение, могут выглядеть как `dbconn0.datasource`, `dbconn0.user` и т.д., а параметры, описывающие выходное соединение - как `dbconn1.datasource`, `dbconn1.user` и т.д. Значения всегда заключаются в двойные кавычки. Файл параметров содержит следующие параметры, и все они могут быть изменены для настройки выполнения потока.

Таблица 3. Параметры выполнения.

Параметр	Назначение
<code>dbconnx.datasource</code>	Имя источника данных (data source name, DSN).
<code>dbconnx.user</code>	Имя пользователя для баз данных с ограниченным доступом.
<code>dbconnx.password</code>	Пароль для баз данных с ограниченным доступом.
<code>dbconnx.epassword</code>	Кодированный пароль для баз данных с ограниченным доступом. Чтобы сгенерировать кодированный пароль, выберите Encode Password (Кодировать пароль) в меню Tools (Инструменты) пользовательского интерфейса IBM SPSS Modeler. Скопируйте кодированный пароль и вставьте его.
<code>dbconnx.password_required</code>	Флаг, указывающий, запрашивать ли пароль для этого соединения с базой данных. (Значение - "Y" (да) или "N" (нет).) Если в момент публикации пароль не задан, автоматически принимает значение "N".
<code>dbconnx.catalog</code>	Каталог, связанный с этим соединением с базой данных.
<code>dbtablex.name</code>	Имя базовой таблицы. <i>Примечание:</i> Порядок полей в новой таблице должен совпадать с порядком полей в первоначально опубликованном потоке.
<code>dbtablex.schema</code>	Имя схемы.
<code>dbtablex.catalog</code>	Каталог, связанный с текущей таблицей.
<code>filex.name</code>	Имя файла данных. <i>Примечание:</i> Порядок полей в новом входном файле должен совпадать с порядком полей в первоначально опубликованном потоке.

Таблица 3. Параметры выполнения (продолжение).

Параметр	Назначение
filex.path	Положение файла (имя каталога).
filex.field_names_included	Для файлов в текстовом формате - флаг, указывающий, что первая строка файла содержит имена полей. (Значение - "Y" (да) или "N" (нет).)
filex.field_separator	Для файлов в текстовом формате - символы, с помощью которых разделяются значения полей.
filex.decimal_separator	Для файлов в текстовом формате - символ, используемый в качестве десятичной запятой.
options.angle_in_radians	Флаг, указывающий, используются ли радианы в качестве единицы измерения в тригонометрических выражениях CLEM. (Значение - "Y" (да) или "N" (нет).)
options.date_2digit_baseline	Определяет столетие для дат, где указаны только две цифры года.
options.date_baseline	Базовый год (всегда 1 января), используемый функциями дат CLEM, работающими с одной датой.
options.time_rollover	Флаг, указывающий, относятся ли к прошлому отрицательные разности времени.
options.decimal_separator	Символ десятичной запятой по умолчанию в текстовых файлах.
options.time_format	Формат времени, используемый функциями времени CLEM, когда строки обрабатываются как время.
options.date_format	Формат даты, используемый функциями дат CLEM, когда строки обрабатываются как даты.
options.timestamp_format	Формат, используемый при чтении полей отметки времени как строк из источников данных ODBC.
paramx.value	Значение параметра потока. <i>Примечание:</i> Имя параметра, его описание и тип хранения включены для справки, но их нельзя изменить. Значение должно быть допустимым для своего типа.

Встраивание IBM SPSS Modeler Solution Publisher в прикладные программы

В дополнение к автономному механизму модуля времени выполнения, IBM SPSS Modeler Solution Publisher содержит библиотеку программирования времени выполнения (CLEMRTL), позволяющую другим программам управлять выполнением потоков IBM SPSS Modeler Solution Publisher. Процедуры CLEMRTL можно вызывать из клиентских программ, написанных на языках C и C++. Для использования CLEMRTL необходимо включить заголовочный файл *clemrtl.h* (доступный в папке *..\installation\clemrtl\include*) и при компоновке указать файл библиотеки, соответствующий вашей платформе разработки, когда выполняется построение прикладной программы. Файл *clemrtl.dll* доступен в папке *..\installation\bin*.

Любой файл с исходным кодом, ссылающийся на библиотечные процедуры, должен включать заголовок *clemrtl.h*. Этот файл заголовка содержит прототипы ANSI C для библиотечных процедур и определяет полезные макроопределения. Он не требует включения каких-либо еще заголовков, за исключением требуемых вашей программой. Для защиты от совпадений имен все имена типов и функций в библиотеке начинаются с *clemrtl_*, а все имена макроопределений - с *CLEMRTL_*.

CLEMRTL создает отчеты с полезной информацией, которую надо каким-то способом вернуть прикладной программе. CLEMRTL предоставляет три механизма работы с такими сообщениями:

- Сообщения могут записываться в файл журнала, заданный при помощи функции *clemrtl_setLogFile()*.

- Прикладная программа может получить подробную информацию о последней ошибке с помощью функции `clemrtl_getErrorDetail()`.
- Прикладная программа может задать собственную процедуру обработки отчетов при помощи функции `clemrtl_setReportHandler()`.

В системе времени выполнения есть каталог локализованных сообщений для отчетов. Любой отчет, передаваемый прикладной программе, содержит локализованную строку сообщения. Кроме того, отчеты включают код отчета, чтобы прикладная программа могла выбирать разные способы интерпретации и представления сообщений. Для получения подробных (локализованных) сообщений прикладная программа должна включать соответствующий файл *messages.cfg* в каталоге *config*. Разные версии *messages.cfg* для разных локалей можно найти в каталоге `<каталог_установки>/config/<локаль>`, где `<каталог_установки>` - каталог, в который вы установили IBM SPSS Modeler Solution Publisher, а `<локаль>` - требуемая локаль. Выберите нужную версию *messages.cfg* и добавьте ее в подкаталог *config* в вашем каталоге проекта.

Система времени выполнения не требует для правильной работы никаких специальных настроек среды или реестра. Зависимые библиотеки должны распространяться вместе с прикладной программой, скомпонованной с CLEMRTL. Эти файлы включены в каталог `<каталог_установки>/bin`.

Требования

В следующей таблице перечислены платформы и компиляторы, протестированные с библиотекой программирования времени выполнения.

Таблица 4. Платформы и компиляторы.

Платформа	Рекомендуемые компиляторы
AIX	IBM XL C/C++ Enterprise Edition V8.0 для AIX
Linux 64-битная	gcc 3.2.3, компилятор C++ Intel 9.1.043
Windows	Visual Studio 2005

Пример прикладной программы IBM SPSS Modeler Solution Publisher

Пример прикладной программы *dlltest.c* находится в папке `\clemrtl\demo` в папке установки IBM SPSS Modeler Solution Publisher. Пример прикладной программы помогает начать встраивание библиотеки времени выполнения в ваши собственные прикладные программы.

Информация, используемая примером прикладной программы, находится в папке `\clemrtl\demo`; инструкции даны в файле *README.txt*. Способ компиляции прикладной программы зависит от платформы, поэтому также предоставляются примеры make-файлов.

Глава 4. Установка модуль времени выполнения IBM SPSS Modeler Solution Publisher

Установка модуля времени выполнения

Установка модуля времени выполнения IBM SPSS Modeler Solution Publisher позволяет выполнять опубликованные потоки. Опубликованные потоки независимы от операционной системы, таким образом, можно выполнить любой опубликованный поток на любой платформе, поддерживаемой модулем времени выполнения.

Функциональная возможность, позволяющая публиковать потоки из IBM SPSS Modeler, устанавливается вместе с IBM SPSS Modeler. Обратите внимание на то, что опубликованные файлы должны заново экспортироваться с каждым выпуском IBM SPSS Modeler. Например, экспортированный с использованием более старой версии файл не будет работать с текущей версией модуль времени выполнения IBM SPSS Modeler Solution Publisher.

Установка в Windows

В следующем разделе описана процедура установки модуля времени выполнения SPSS Modeler Solution Publisher в операционной системе Windows.

1. Запустите установку IBM SPSS Modeler Solution Publisher.
2. В меню установки выберите **Установить модуль времени выполнения IBM SPSS Modeler Solution Publisher** или **Установить модуль времени выполнения IBM SPSS Modeler Solution Publisher для Win64** (Установить модуль времени выполнения IBM SPSS Modeler Solution Publisher для Win64) в зависимости от системы, на которую производится установка.
3. В мастере по установке нажмите кнопку **Далее**, чтобы начать установку.
4. Следуйте инструкциям на экране. Чтобы продолжить, нажмите кнопку **Далее**.
5. Когда все опции будут заданы, можно начинать установку. Нажмите кнопку **Установить**, чтобы начать перенос файлов.
6. Когда все файлы будут установлены, нажмите кнопку **Готово**.

Установка в UNIX

В следующих разделах описаны дополнительные требования к системе и процедура установки модуля времени выполнения IBM SPSS Modeler Solution Publisher в операционной системе UNIX.

Дополнительные требования для UNIX

Вы должны убедиться в том, что ограничения ядра в системе допускают работу IBM SPSS Modeler Server. В частности, важно отсутствие ограничений для данных, памяти и файлов, и они должны быть сняты в среде IBM SPSS Modeler Server. Для этого:

1. Добавьте следующие команды в *modelersrv.sh*:

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

Кроме того, задайте предел стека равным максимуму для вашей системы (*ulimit -s XXXX*), например:

```
ulimit -s 64000
```

2. Перезапустите IBM SPSS Modeler Server.

Также понадобится утилита сжатия файлов *gzip* и *GNU cpio*, которые должны быть установлены и включены в переменную PATH, чтобы программа установки смогла распаковать файлы установки. Кроме того, на компьютере, где работает SPSS Modeler Server, надо установить локаль EN_US.UTF-8.

Процедура установки в UNIX

Эти инструкции по установке относятся к модулю времени выполнения IBM SPSS Modeler Solution Publisher для UNIX. Вам потребуются разрешения на чтение и запись для каталога установки назначения, поэтому войдите в систему под учетной записью с достаточными полномочиями.

Примечание: Для установки вы должны войти в систему как пользователь *root*.

1. Из пакета установки откройте каталог *modelrun*.
2. Перейдите в соответствующий каталог платформы.
3. Запустите сценарий установки *.bin*. Например:

```
./modelersolutionpublisherhpia64.bin -i console
```
4. Детали применения выводятся. Нажмите клавишу Enter для продолжения.
5. Выводится информация, касающаяся лицензионного соглашения. Прочтите текст лицензионного соглашения, наберите 1, чтобы его принять, и нажмите Enter для продолжения.
6. Выводится список доступных языков. Введите число, соответствующее языковой версии, которую нужно установить, и нажмите клавишу Enter, чтобы продолжить.
7. Вам будет предложено указать место установки. Для использования каталога по умолчанию, */usr/IBM/SPSS/ModelerSolutionPublisher<nn>* (где *<nn>* - номер версии), нажмите клавишу Enter. Модуль времени выполнения будет установлен в указанный каталог.
8. Вам будет предложено подтвердить заданное место установки. В случае, если оно указано верно, наберите у и нажмите Enter.
9. Чтобы дать возможность проверить информацию, введенную вами до настоящего момента, выводится предустановочная сводка. Нажмите клавишу Enter для продолжения.
10. Будет выведено сообщение, извещающее о том, что программа установки готова к выполнению. Нажмите клавишу Enter для продолжения.
11. Во время выполнения программы выводится индикатор хода ее выполнения. Когда установка будет завершена, нажмите Enter для выхода из программы установки.

Конфигурирование ODBC в UNIX

В системах UNIX по умолчанию в модуль времени выполнения IBM SPSS Modeler Solution Publisher не сконфигурировано использование ODBC через менеджер драйверов DataDirect. Чтобы настроить UNIX для загрузки менеджера драйверов DataDirect, введите следующие команды (где *каталог_установки_sp* - каталог установки Solution Publisher Runtime):

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

Устранение неполадок при установке

Цифровая подпись установочных файлов недействительна

В цифровой подписи продуктов IBM SPSS Modeler используются сертификаты IBM. В некоторых случаях при попытке установить продукты SPSS Modeler будут возникать ошибки, которые перечислены ниже.

Ошибка 1330. Невозможно установить требуемый файл из-за недействительной цифровой подписи CAB-файла...

Для пользователей всех ОС Windows

Это сообщение будет выведено при попытке установить SPSS Modeler на компьютере без подключения к Интернету и без установленных действительных сертификатов. Чтобы устранить эту проблему, воспользуйтесь процедурой, которая приведена ниже.

1. Чтобы принять сообщение, нажмите кнопку **ОК**.
2. Чтобы выйти из программы установки, нажмите кнопку **Отмена**.
3. Если на компьютере установки отсутствует подключение к Интернету, выполните следующее действие на компьютере с подключением к Интернету и скопируйте файл `.cer` на первый компьютер.
4. Перейдите по адресу <https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&id=SO5624> и следуйте инструкциям, чтобы скачать **корневой сертификат VeriSign Class 3 Primary Certification Authority - G5**. Сохраните его как файл `.cer`.
5. Щелкните дважды по файлу `.cer`.
6. На вкладке Общие свойства щелкните **Установить сертификат**.
7. Следуйте инструкциям мастера импорта сертификатов, используя параметры по умолчанию. По окончании нажмите кнопку **Завершить**.
8. Повторите установку.

Запуск модуля времени выполнения

После установки модуля времени выполнения с его помощью можно выполнять потоки, опубликованные из IBM SPSS Modeler с использованием любого узла экспорта. Дополнительную информацию смотрите в разделе “Публикация потоков” на стр. 8. Модуль времени выполнения запускается из командной строки с опциями, указывающими имя опубликованного файла образа и (необязательно) файла параметров, которые надо запустить.

Для запуска модуля времени выполнения введите в командной строке следующую команду:

```
modelerrun -p <файл-параметров> -o <опции> <файл-образа>
```

Описание параметров команды дано в следующей таблице.

Таблица 5. Параметры команды модуля времени выполнения

Параметр команды	Описание
<файл-параметров>	имя файла опубликованного файла параметров (необязательно).
<опции>	список пар вида опция=значение через запятую, указывающих опции выполнения для модуля времени выполнения. Примечание: После запятой не должно быть пробельных символов.
<файл-образа>	имя файла опубликованного файла образа (<i>*.pim</i>) для выполнения.

Примечание: При использовании языков с многобайтной кодировкой, например, китайского или японского, в среде UNIX, необходимо указать используемые коды языка. В следующем примере показаны коды языка для упрощенного китайского:

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

где `-p` - положение файлов `pim` и `par`.

Дополнительную информацию смотрите в разделе Глава 3, “Выполнение опубликованных потоков”, на стр. 11.

Изменение каталога temp

Некоторые операции модуля времени выполнения IBM SPSS Modeler Solution Publisher могут потребовать создания временных файлов. По умолчанию модуль времени выполнения использует для создания временных файлов временный каталог системы. Изменить положение временного каталога можно при помощи следующих действий.

Примечание: очень важно использовать правильный синтаксис, обратите особое внимание на пробелы и путь к каталогу.

Допустим, что вы используете модуль времени выполнения IBM SPSS Modeler Solution Publisher с опубликованными файлами *webtest.pim* и *webtest.par*. В этой ситуации вы хотите изменить положение временных файлов, задав каталог *C:\published\temp*. Чтобы изменить положение временного каталога, используемого исполняемым файлом:

1. Из каталога, в котором сохранены файлы *.pim* и *.par*, выполните команду:

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

Это заставит модуль времени выполнения использовать "*C:\published\temp*" в качестве временного каталога только для этого конкретного процесса.

Деинсталляция модуля времени выполнения IBM SPSS Modeler Solution Publisher

В этом разделе описана процедура деинсталляции модуля времени выполнения IBM SPSS Modeler Solution Publisher.

Процедура деинсталляции в Windows

1. На панели управления Windows откройте **Установка и удаление программ**.
2. Выберите в списке **IBM SPSS Modeler Solution Publisher <nn>**.
3. Нажмите кнопку **Удалить**.
4. Следуйте инструкциям на экране, чтобы удалить модуль времени выполнения.

Примечание: Для удаления программ вы должны быть зарегистрированы на вашем компьютере с привилегиями администратора.

Процедура деинсталляции в UNIX

Удалите каталог установки вместе со всеми установленными файлами.

Глава 5. API встроенного предсказательного моделирования IBM SPSS

API IBM SPSS EPM (Embedded Predictive Modeling - встроенное предсказательное моделирование) определяет простой интерфейс Java, позволяющий интегрировать соответствующие проекту IBM SPSS Modeler артефакты с прикладными программами методом встраивания. Полная документация в формате Javadoc входит в состав IBM SPSS Modeler Solution Publisher для кодировщиков, использующих эту библиотеку.

При использовании API Embedded Predictive Modeling (EPM) следует учитывать следующие ограничения:

- **Требуются уникальные метки исходного и конечного узла.** API EPM используют метки узлов. Хотя эти метки всегда соответствуют контексту исходного или конечного узла, они могут быть неоднозначны.
- **Надузлы как исходные или конечные узлы не поддерживаются.** В целях обеспечения общего для надузлов источника во всех операциях надузлы в качестве исходных узлов или конечных узлов не поддерживаются.
- **Ограничения на имена входных полей.** В именах полей нельзя использовать некоторые символы.

Обзор функциональных возможностей этого API смотрите в следующем описании. Дополнительные подробности смотрите в поставляемой с продуктом документации Javadoc.

Встроенный скоринг предсказательной модели

Использование: Используется прикладной программой для генерирования предсказательной аналитики по требованию по мере поступления новых данных. Прикладная программа должна обрабатывать обновленные модели. Все исходные узлы (так же, как и конечный узел ветви скоринга) заменяются в этом режиме скоринга входными и выходными данными прикладной программы.

Действие: Подготавливает ветвь скоринга, проектируемую в потоке IBM SPSS Modeler, один раз для повторяющихся вызовов, которые быстро и эффективно генерируют предсказательную аналитику из входных данных, управляемых прикладной программой.

Обновление предсказательной модели

Использование: Выполняется начальное обучение алгоритмов модели в проектном решении потока при помощи хронологических данных. Кроме того, выполняется периодическое "обновление" сконфигурированных алгоритмов предсказательной модели с учетом большинства текущих данных хронологии.

Действие: Запускает все узлы построителя предсказательной модели в потоке IBM SPSS Modeler, который обучает алгоритм предсказательной модели как сконфигурированный при помощи технологий исследования данных. Обновляет слепок применителя модели, связанный с каждым узлом построителя моделей. Параметры потока и свойства исходного/конечного узла могут использоваться для реализации определенной степени динамики процесса.

Оценка предсказательной модели

Использование: Используется прикладной программой для сообщения об оценке точности, достоверности или ином качественном анализе обучающей предсказательной модели в соответствии с проектным решением потока IBM SPSS Modeler. Эта информация обычно используется для управления переводом только что обновленной предсказательной модели в режим использования для скоринга прикладной программы. Основная причина, по которой операция обновления модели возвращает новый объект `PredictiveModel` - это включение поддержки сравнения данных оценки только что обновленной модели и исходной модели.

Действие: Запускает все узлы построителя документов (в первую очередь неинтерактивные узлы вывода на палитрах вывода и диаграмм IBM SPSS Modeler) в потоке IBM SPSS Modeler, возвращающем итоговые документы. Параметры потока и свойства исходного/конечного узла могут использоваться для реализации определенной степени динамики процесса.

Выполнение потока предсказательной модели

Использование: Все случаи использования, не попадающие в "обновление", "оценку" и два режима скоринга, должны обрабатываться посредством выполнения потока.

Действие: Выполнение, управляемое опциями потока IBM SPSS Modeler. Либо в режиме "все конечные узлы", аналогичном нажатию кнопки с большой зеленой стрелкой на клиенте IBM SPSS Modeler, либо как закодированное в сценарии уровня потока. Параметры потока и свойства исходного/конечного узла могут использоваться для реализации определенной степени динамики процесса.

Внешние данные скоринга предсказательной модели

Использование: Процесс, обычно называемый "пакетным" из-за способа, которым обрабатывается полный пакет входных данных.

Действие: Запускает ветвь скоринга в соответствии с проектным решением. Входные данные с одного или нескольких сконфигурированных узлов источников данных и сгенерированная предсказательная аналитика сохраняются под управлением конечного узла. Параметры потока и свойства исходного/конечного узла могут использоваться для реализации определенной степени динамики процесса.

Параметры потока предсказательной модели и спецификация свойств исходного и конечного узла

Использование: Принятый способ определения параметров, на которые ссылаются выражения в ветвях потока. Еще одна точка управления - это изменение определенных свойств ключей исходного или конечного узла с целью изменения используемого пути файлов и так далее.

Действие: Могут быть изменены определенные аспекты выполнения потока IBM SPSS Modeler прикладной программой.

Примечание: Изменения в параметры должны быть внесены перед созданием PredictiveScorer (смотрите описание встроенного скоринга выше), а свойства исходного и конечного узла не имеют для этого объекта никакого значения.

Глава 6. Справочник по API библиотеки времени выполнения IBM SPSS Modeler Solution Publisher

Введение

API CLEMRTL позволяет управлять выполнением опубликованных потоков из вашей прикладной программы. В этом разделе описаны доступные функции API.

Основные положения, которые следует принимать во внимание при использовании этого API:

- API использует компоновку языка C для максимальной совместимости, однако у библиотеки есть зависимости C++. На некоторых платформах это означает, что его можно использовать только с компоновщиком, поддерживающим C++.
- `clemrtl_initialise_ext()` должна быть вызвана до использования любых других функций в библиотеке.
- Для идентификатора образа используется тип `clemrtl_image_handle`.
- Каждая функция возвращает код индикатора состояния. Значения индикатора состояния описаны в следующей таблице.

Таблица 6. Значения индикатора состояния

Результат	Описание
CLEMRTL_OK	Успех
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны
CLEMRTL_ERROR	Неудачное завершение; дополнительная информация об ошибке доступна при помощи функции <code>clemrtl_getErrorDetail()</code>

Обзор процессов API

Общий обзор программ, использующих API:

1. Инициализация библиотеки использует `clemrtl_initialise_ext()`.
2. Опции проверки и изменения выполнения используют `clemrtl_getOption()` и `clemrtl_setOption()`.
3. Открытие образа использует `clemrtl_openImage()` и получает хэндл образа.
4. Проверка и изменение параметров образа используют `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` и `clemrtl_setParameter()`.
5. Выполнение образа использует `clemrtl_execute()`. Если тот же образ выполняется несколько раз без изменения параметров, используйте сначала `clemrtl_prepare()`.
6. Закрытие образа использует `clemrtl_closeImage()`.
7. Для отмены начатого выполнения используйте `clemrtl_interrupt()`.
8. Для получения информации о последней ошибке используйте `clemrtl_getErrorDetail()`. Для получения сообщений об ошибках и других диагностических сообщений по мере их поступления используйте `clemrtl_setReportHandler()`.

Эта библиотека предназначена для многопоточного использования при следующих условиях:

- Глобальные функции, не принимающие хэндл образа как аргумент, не могут вызываться одновременно.
- Локальные для образа функции, принимающие хэндл образа как аргумент, нельзя вызывать одновременно при применении к одному и тому же хэндлу образа, за исключением `clemrtl_interrupt()`.
- Локальные для образа функции можно вызывать одновременно при применении к разным хэндлам образов.

В частности, это означает, что отдельные потоки могут подготавливать и выполнять образы одновременно, при условии, что они используют разные хэндлы образов. Разные хэндлы образов могут ссылаться на один и тот же образ (`clemrtl_openImage()` создает новый хэндл при каждом новом вызове, даже для одного и того же образа), но при выполнении нескольких экземпляров одного образа не забывайте изменять параметры образа для правильного перенаправления вывода. Если два образа, выполняемые одновременно, пытаются записывать данные в один и тот же выходной файл или таблицу базы данных, результаты могут быть непредсказуемыми.

Функции API

Далее приведен полный список функций, представленных в API.

initialise

Эта функция объявлена устаревшей и эквивалентна следующей

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

Новые программы должны вызвать *initialise_ext*.

initialise_ext

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

Инициализирует модуль времени выполнения IBM SPSS Modeler Solution Publisher. *Эту функцию следует вызывать до вызова любых других функций API.*

Таблица 7. Параметры *initialise_ext*

Параметр	Описание
flags	Управляет некоторыми аспектами процесса инициализации. Это значение формируется как побитовое ИЛИ флагов, описанных ниже.
arg_count	Число дополнительных параметров инициализации. Это число не должно превышать длину массива args.
args	Дополнительные параметры инициализации. Параметры описываются следующим типом: <pre>typedef struct _clemrtl_init_arg{ const char* name; const char* value; } clemrtl_init_arg;</pre>

Поле name - имя свойства конфигурации, поле value - его значение. Поддерживаемые свойства конфигурации приведены в следующей таблице.

Таблица 8. Свойства конфигурации

Имя	Значение
installation_directory	Полный путь к папке установки IBM SPSS Modeler Solution Publisher. Папка установки - это папка, в которой находятся папки bin и config.

Возвращает один из кодов ошибок, показанных в таблице ниже.

Таблица 9. Коды ошибок

Результат	Описание
CLEMRTL_OK	Успех.

Таблица 9. Коды ошибок (продолжение)

Результат	Описание
CLEMRTL_FAIL	Неудачная инициализация; подробности недоступны.

Параметр `flags` позволяет управлять некоторыми аспектами процесса инициализации. Значение 0 (или `CLEMRTL_INIT_DEFAULTS`) задает следующее поведение по умолчанию:

- Часовой пояс по умолчанию задается путем вызова `tzset()`.
- Локаль `LC_NUMERIC` устанавливается в "C".
- Устанавливается глобальный новый обработчик, генерирующий исключительную ситуацию в случае исчерпания памяти.

Это поведение можно изменить путем указания сочетаний флагов, показанных в следующей таблице.

Таблица 10. Флаги

Флаг	Описание
<code>CLEMRTL_INIT_NO_TZ</code>	Не задавать часовой пояс.
<code>CLEMRTL_INIT_NO_LOCALE</code>	Не менять локаль.
<code>CLEMRTL_INIT_NO_NEW_HANDLER</code>	Не устанавливать новый обработчик.
<code>CLEMRTL_INIT_LOCAL_NEW_HANDLER</code>	Локализовать новый обработчик внутри каждого вызова API.

Для правильного преобразования между числами и строками библиотека требует, чтобы для локали `LC_NUMERIC` было задано значение "C" (числовой формат IBM SPSS Modeler использует правила локали "C"). Это включает преобразование чисел, считываемых из текстовых файлов и записываемых туда.

- Определите `NO_LOCALE`, если ваша прикладная программа зависит от локали `LC_NUMERIC` и не требует преобразования между числами и строками.
- `NO_NEW_HANDLER` и `LOCAL_NEW_HANDLER` - взаимоисключающие флаги. Библиотека требует, чтобы операция `new` генерировала исключительную ситуацию, когда память исчерпана.
- Укажите `NO_NEW_HANDLER`, если ваша прикладная программа устанавливает свой собственный новый обработчик или если вы уверены, что ваша система компилятора/библиотеки времени выполнения генерирует исключительную ситуацию по умолчанию.

Указывайте `LOCAL_NEW_HANDLER`, только если ваша прикладная программа требует, чтобы операция `new` возвращала нулевой указатель при исчерпании памяти. *Примечание:* Это небезопасно использовать в многопоточной среде.

getOption

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

Возвращает значение опции выполнения.

Таблица 11. Параметры `getOption`

Параметр	Описание
<code>name</code>	Имя опции.
<code>value</code>	Буфер для получения значения опции.
<code>value_size</code>	Размер буфера для значения.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 12. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если имя опции не совпадает ни с одним из имен опций выполнения, перечисленных выше — например, `memory_usage`, — или если буфер значения недостаточно велик для получения значения параметра, включая завершающий нулевой символ. Значения опций всегда являются строками, даже если интерпретируются как числа. Дополнительную информацию смотрите в разделе Глава 3, “Выполнение опубликованных потоков”, на стр. 11.

setOption

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

Задаёт значение опции выполнения.

Таблица 13. Параметры `setOption`

Параметр	Описание
<code>name</code>	Имя опции.
<code>value</code>	Значение опции.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 14. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если имя опции не совпадает ни с одним из имен опций выполнения, перечисленных выше — например, `memory_usage`. Значения опций всегда являются строками, даже если интерпретируются как числа.

setLogFile

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

Перенаправляет сообщения журнала от всех впоследствии открытых образов.

Таблица 15. Параметры *setLogFile*

Параметр	Описание
mode	Режим записи в журнал должен быть одним из следующих значений: CLEMRTL_NULL_LOG — подавлять сообщения. CLEMRTL_STDERROR_LOG — отправлять сообщения на стандартный вывод ошибок. CLEMRTL_FILE_LOG — отправлять сообщения в файл, имя которого задано параметром <i>log_file</i> .
log_file	Строка, содержащая имя файла.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 16. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если указан неизвестный режим или если затребована запись журнала в файл, но имя файла не указано или недопустимо.

openImage

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

Открывает опубликованный образ потока.

Таблица 17. Параметры *openImage*

Параметр	Описание
image_file	Имя файла образа.
param_file	Имя файла параметров.
handle	Получает хэндл образа.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 18. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Если отдельный файл параметров не требуется, в качестве файла параметров может быть передано значение NULL. Операция завершается неудачно, если не указан файл образа, если одно из имен файлов недопустимо или если не удастся прочитать содержимое файла. Если операция завершается успешно, возвращенный хэндл образа можно использовать для идентификации экземпляра образа в последующих вызовах API.

closeImage

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

Закрывает хэндл образа и освобождает его ресурсы.

Таблица 19. Параметры *closeImage*

Параметр	Описание
handle	Хэндл образа.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 20. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если с этим образом работает уже выполняемый вызов API, в частности, если образ все еще выполняется. Если операция завершается успешно, образ закрывается, и этот хэндл становится нельзя использовать в любых последующих вызовах API, за исключением другого вызова `clemrtl_closeImage()`, который не произведет никакого эффекта.

enumerateParameters

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

Применяет процедуру обратного вызова для каждого имени и значения параметра образа.

Таблица 21. Параметры *enumerateParameters*

Параметр	Описание
handle	Хэндл образа.
proc	Процедура обратного вызова для параметра.
data	Пользовательские данные для обратного вызова.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 22. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Процедура обратного вызова имеет следующий тип:

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


Таблица 23. Параметры процедуры обратного вызова

Параметр	Описание
data	Пользовательские данные, переданные функции <code>clemrtl_enumerateParameters()</code> .
name	Имя параметра.
value	Значение параметра.

Процедура применяется ровно по одному разу к каждому параметру образа в произвольном порядке.

getParameter

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

Получает значение параметра образа.

Таблица 24. Параметры `getParameter`

Параметр	Описание
handle	Хэндл образа.
name	Имя параметра.
value	Буфер для получения значения параметра.
value_size	Размер буфера значения.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 25. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если имя параметра не совпадает ни с одним из имен параметров в образе, или если буфер значения недостаточно велик для получения значения параметра, включая завершающий нулевой символ. Для имен параметров используется описанный выше формат имя.атрибут (например, `file0.name`), а значения параметров всегда являются строками, даже если интерпретируются как числа.

setParameter

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

Задаёт значение параметра образа.

Таблица 26. Параметры `setParameter`

Параметр	Описание
handle	Хэндл образа.

Таблица 26. Параметры *setParameter* (продолжение)

Параметр	Описание
name	Имя параметра.
value	Значение параметра.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 27. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если хэндл образа недопустим или если имя параметра не совпадает ни с одним из имен параметров в образе. Для имен параметров используется описанный выше формат имя.атрибут (например, file0.name), а значения параметров всегда являются строками, даже если интерпретируются как числа.

getFieldCount

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

Возвращает число полей в источнике ввода или в назначении вывода.

Таблица 28. Параметры *getFieldCount*

Параметр	Описание
handle	Хэндл образа.
key	Имя ввода или вывода, которые нужно исследовать, как они используются в файле параметров. Этот ключ может ссылаться на файл или базу данных.
field_count	Возвращает число полей.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 29. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

getFieldTypes

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

Возвращает типы полей для источника ввода или для назначения вывода.

Таблица 30. Параметры `getFieldTypes`

Параметр	Описание
<code>handle</code>	Хэнгл образа.
<code>key</code>	Имя ввода или вывода для изучения, как они используются в файле параметров. Этот ключ может ссылаться на файл или базу данных.
<code>field_count</code>	Число полей для изучения. Это должно быть число, не превышающее длины массива <code>field_types</code> .
<code>field_types</code>	Массив длиной не менее <code>field_count</code> , получающий типы полей. В массив копируются типы первых <code>field_count</code> полей в источнике или назначении. Если <code>field_count</code> больше фактического числа полей, содержимое лишних элементов массива не определено. Значения для типов полей описаны ниже в таблице Типы данных.

Типы данных

Каждый элемент массива `field_types` должен быть одним из типов данных, показанных в следующей таблице.

Таблица 31. Типы данных

Тип	Интерпретация	Обычное объявление на 'C'
STRING	Строка символов в кодировке UTF-8, заканчивающаяся нулевым байтом.	<code>const char*</code>
INTEGER	32-битное целое число со знаком.	<code>int</code>
LONG	64-битное целое число со знаком.	<code>long long</code>
REAL	64-битное число с плавающей точкой.	<code>double</code>
TIME	64-битное целое число со знаком (число секунд, прошедших с полуночи).	<code>long long</code>
DATE	64-битное целое число со знаком (число секунд, прошедших с полуночи 01.01.1970).	<code>long long</code>
TIMESTAMP	64-битное целое число со знаком (число секунд, прошедших с полуночи 01.01.1970).	<code>long long</code>

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 32. Коды состояния

Результат	Описание
<code>CLEMRTL_OK</code>	Успех.
<code>CLEMRTL_FAIL</code>	Неудачное завершение; подробности недоступны.
<code>CLEMRTL_ERROR</code>	Неудачное завершение; подробности доступны.

setAlternativeInput

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

Заменяет файловый источник ввода на альтернативный источник ввода.

Таблица 33. Параметры *setAlternativeInput*

Параметр	Описание
handle	Хэндл образа.
key	Имя первоначального источника ввода, как оно указано в файле параметров. Источник ввода должен быть файлом, так что ключ всегда будет иметь вид “fileN”, где целое число $N \geq 0$.
field_count	Число полей в вводе. Это значение должно в точности соответствовать числу полей в первоначальном файловом источнике ввода, иначе вызов завершится неудачно.
field_types	Массив типов, длина которого должна быть как минимум равна <code>field_count</code> . Значение <code>field_types[i]</code> - это тип i-го поля, оно должно быть одним из значений, указанных ниже в таблице Типы данных. Типы должны быть совместимыми с типами первоначального файлового источника данных, иначе вызов завершится неудачно.
iterator	<p>Функция, порождающая альтернативные входные данные. Эта функция применяется к ее параметру следующим образом:</p> <pre>void** row = iterator(arg);</pre> <p>Функция вызывается при выполнении (внутри вызова <code>clermrtl_execute</code>) по одному разу для каждой входной записи. Если функция возвращает NULL, это означает конец ввода; в этом случае функция больше не вызывается и выполнение будет в конечном счете завершено. В противном случае результат должен быть массивом данных длиной по меньшей мере <code>field_count</code>, где <code>row[i]</code> содержит значение i-го поля. Значение поля может быть равно NULL, иначе оно должно быть указателем на данные, тип которых определяется соответствующим элементом <code>field_types[i]</code>. Указатели должны быть действительны до следующего вызова итератора или до конца выполнения, если выполнение будет преждевременно завершено.</p>
arg	Непрозрачный аргумент, который передается итератору при каждом вызове.

Типы данных

Каждый элемент массива `field_types` должен быть одним из типов, список которых ограничен. Дополнительную информацию смотрите в разделе “getFieldTypes” на стр. 28.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 34. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если типы полей несовместимы с типами полей в первоначальном файловом источнике данных, или если число полей не равно числу полей в первоначальном файловом источнике данных.

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);
```

Заменяет файловое назначение вывода альтернативным назначением вывода.

Таблица 35. Параметры `setAlternativeOutput`

Параметр	Описание
handle	Хэндл образа.
key	Имя первоначального назначения вывода, как оно указано в файле параметров. Назначение вывода должно быть файлом, так что ключ всегда будет иметь вид “fileN”, где целое число $N \geq 0$.
field_count	Число полей в выводе. Это значение должно в точности соответствовать числу полей в первоначальном файловом назначении вывода, иначе вызов завершится неудачно.
field_types	Массив типов, длина которого должна быть как минимум равна <code>field_count</code> . Значение <code>field_types[i]</code> - это тип <i>i</i> -го поля, оно должно быть одним из значений, указанных ниже в таблице Типы данных. Типы должны быть совместимыми с типами первоначального файлового назначения вывода, иначе вызов завершится неудачно.
iterator	Функция, принимающая вывод образа. Эта функция применяется к ее параметру и к строке данных следующим образом: <code>iterator(arg, row);</code> Функция вызывается при выполнении (внутри вызова <code>clemrtl_execute</code>) по одному разу для каждой строки результата, сгенерированной образом. Значение <code>row</code> , равное <code>NULL</code> , обозначает конец вывода; после него функция больше не должна вызываться; прикладные программы не должны полагаться на этот заключительный вывод и должны выполнять сброс и закрытие любых внешних ресурсов и т.д. по завершении выполнения. Иначе параметр <code>row</code> должен быть массивом данных длиной по меньшей мере <code>field_count</code> , где <code>row[i]</code> содержит значение <i>i</i> -го результата. Значение поля может быть равно <code>NULL</code> , иначе оно должно быть указателем на данные, тип которых определяется соответствующим элементом <code>field_types[i]</code> . Функция должна скопировать любые нужные ей значения данных, так как по возвращении из вызова сохранение содержимого этой памяти не гарантируется.
arg	Непрозрачный аргумент, который передается итератору при каждом вызове.

Типы данных

Каждый элемент массива `field_types` должен быть одним из типов из ограниченного списка. Дополнительную информацию смотрите в разделе “`getFieldTypes`” на стр. 28.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 36. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_FAIL	Неудачное завершение; подробности недоступны.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если типы полей несовместимы с типами полей в первоначальном файлом назначении вывода, или если число полей не равно числу полей в первоначальном файлом назначении вывода.

execute

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

Выполняет образ.

Таблица 37. Параметры execute

Параметр	Описание
handle	Хэндл образа.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 38. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Если образ не был подготовлен, он будет сначала подготовлен с использованием текущих значений параметров. Операция завершается неудачно, если хэндл образа недопустим или если при подготовке или выполнении произошла ошибка. Этот вызов не возвращает управление до завершения выполнения.

prepare

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

Подготавливает образ для выполнения.

Таблица 39. Параметры prepare

Параметр	Описание
handle	Хэндл образа.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 40. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Операция завершается неудачно, если хэндл образа недопустим или если при подготовке произошла ошибка.

Перед выполнением образа его необходимо подготовить. Подготовка образа фиксирует значения параметров в образе. `clemrtl_execute()` автоматически подготавливает образ при каждом вызове, если он еще не подготовлен; использовать `clemrtl_prepare()` для подготовки образа полезно, если образ нужно выполнить несколько раз с одними и теми же значениями параметров; этот вызов может значительно улучшить производительность. После подготовки образа последующие изменения значений параметров игнорируются; чтобы обновить значения параметров в образе, следует снова вызвать `clemrtl_prepare()`.

interrupt

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

Прерывает выполнение образа.

Таблица 41. Параметры *interrupt*

Параметр	Описание
handle	Хэндл образа.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 42. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Эта операция не имеет никакого эффекта, если хэндл образа недопустим или если образ не выполняется.

Эту функцию безопасно вызывать одновременно с другим вызовом API с тем же хэндлом образа.

getErrorDetail

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

Получает подробную информацию о последней ошибке, произошедшей в образе.

Таблица 43. Параметры *getErrorDetail*

Параметр	Описание
handle	Хэндл образа.
severity	Получает код серьезности в виде одного символа: I — информация W — предупреждение E — ошибка X — системная ошибка
code	Получает номер ошибки.
text	Буфер для получения текста сообщения.
text_size	Размер текстового буфера.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 44. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Если хэндл образа недопустим, результаты будут ненадежны. При необходимости текст сообщения усекается для помещения в текстовый буфер.

Если вызов API завершился неудачно с кодом возврата CLEMRTL_ERROR и эта ошибка не связана с конкретным хэндлом образа, можно передать 0 в качестве хэндла образа для получения подробной информации о последней ошибке, не специфичной для конкретного образа, но эта возможность ненадежна в многопоточной среде.

setReportHandler

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

Устанавливает процедуру обратного вызова отчета для образа.

Таблица 45. Параметры setReportHandler

Параметр	Описание
handle	Хэндл образа.
proc	Процедура обратного вызова отчета.
data	Пользовательские данные для обратного вызова.

Возвращает один из кодов состояния, показанных в таблице ниже.

Таблица 46. Коды состояния

Результат	Описание
CLEMRTL_OK	Успех.
CLEMRTL_ERROR	Неудачное завершение; подробности доступны.

Процедура обратного вызова имеет следующий тип:

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

Таблица 47. Параметры процедуры обратного вызова

Параметр	Описание
data	Пользовательские данные, переданные функции clemrtl_setReportHandler().

Таблица 47. Параметры процедуры обратного вызова (продолжение)

Параметр	Описание
severity	Код серьезности в виде одного символа: I — информация W — предупреждение E — ошибка X — системная ошибка
code	Номер сообщения.
text	Текст сообщения.

Процедура применяется к каждому поступающему сообщению. Задание для процедуры значения NULL удаляет любой существующий обработчик, установленный для образа.

Уведомления

Эта информация относится к продуктам и сервису, предлагаемым в США. Этот материал может быть доступен от IBM на других языках. Однако для его получения может понадобиться приобрести продукт или версию продукта на нужном языке.

IBM может не предоставлять в других странах продукты, услуги и аппаратные средства, описанные в данном документе. За информацией о продуктах и услугах, предоставляемых в вашей стране, обращайтесь к местному представителю IBM. Ссылки на продукты, программы или услуги IBM не означают и не предполагают, что можно использовать только указанные продукты, программы или услуги IBM. Разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако ответственность за оценку и проверку работы любого продукта, программы или сервиса, не произведенного корпорацией IBM, лежит на пользователе.

IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данного документа. Предъявление данного документа не предоставляет какую-либо лицензию на эти патенты. Вы можете послать письменный запрос о лицензии по адресу:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

По поводу лицензий, связанных с использованием наборов двухбайтных символов (DBCS), обращайтесь в отдел интеллектуальной собственности IBM в вашей стране или направьте запрос в письменной форме по адресу:

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

INTERNATIONAL BUSINESS MACHINES CORPORATION ПРЕДСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, КАК ЯВНЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ, ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ СОБЛЮДЕНИЯ ЧЬИХ-ЛИБО АВТОРСКИХ ПРАВ, ВОЗМОЖНОСТИ КОММЕРЧЕСКОГО ИСПОЛЬЗОВАНИЯ ИЛИ ПРИГОДНОСТИ ДЛЯ КАКИХ-ЛИБО ЦЕЛЕЙ И СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ. В некоторых странах для ряда сделок не допускается отказ от явных или предполагаемых гарантий; в таком случае данное положение к вам не относится.

Эта информация может содержать технические неточности и типографские ошибки. В представленную здесь информацию периодически вносятся изменения; эти изменения будут включаться в новые издания данной публикации. Фирма IBM может в любое время без уведомления вносить изменения и усовершенствования в продукты и программы, описанные в этой публикации.

Любые ссылки в этой публикации на сайты, не принадлежащие IBM, приведены только для удобства и никоим образом не означают их поддержки. Материалы на этих сайтах не входят в число материалов по данному продукту IBM, и весь риск пользования этими сайтами несете вы сами.

Любую предоставленную вами информацию IBM может использовать или распространять любым способом, какой сочтет нужным, не беря на себя никаких обязательств по отношению к вам.

Если обладателю лицензии на данную программу понадобятся сведения о возможности: (i) обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) совместного использования таких данных, он может обратиться по адресу:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Такая информация может быть доступна при соответствующих условиях и соглашениях, включая в некоторых случаях взимание платы.

Описанную в данном документе лицензионную программу и все прилагаемые к ней лицензированные материалы IBM предоставляет на основе положений Соглашения между IBM и Заказчиком, Международного Соглашения о Лицензиях на Программы IBM или любого эквивалентного соглашения между IBM и заказчиком.

Данные производительности и примеры клиентов представлены только для иллюстрации. Фактическая производительность зависит от конкретной конфигурации и условий работы.

Информация о продуктах других компаний (не IBM) получена от поставщиков этих продуктов, из их опубликованных объявлений или из иных общедоступных источников. IBM не производила тестирование этих продуктов и никак не может подтвердить информацию о их точности работы и совместимости, а также прочие заявления относительно продуктов других компаний (не IBM). Вопросы о возможностях продуктов других компаний (не IBM) следует направлять поставщикам этих продуктов.

Все утверждения о будущих планах и намерениях IBM могут быть изменены или отменены без уведомлений, и описывают исключительно цели фирмы.

Эти сведения содержат примеры данных и отчетов, используемых в повседневных деловых операциях. Чтобы проиллюстрировать их настолько полно, насколько это возможно, данные примеры включают имена индивидуумов, названия компаний, брендов и продуктов. Все эти имена являются вымышленными и любое их сходство с реальными именами и адресами предприятий является случайным.

Товарные знаки

IBM, логотип IBM, и ibm.com являются товарными знаками или зарегистрированными товарными знаками компании International Business Machines Corp., зарегистрированными во многих странах мира. Прочие наименования продуктов и услуг могут быть товарными знаками, принадлежащими IBM или другим компаниям. Текущий список товарных знаков IBM смотрите на веб-сайте "Copyright and trademark information" (Информация об авторских правах и товарных знаках) по адресу www.ibm.com/legal/copytrade.shtml.

Adobe, логотип Adobe, PostScript и логотип PostScript являются либо зарегистрированными товарными знаками, либо товарными знаками корпорации Adobe Systems в Соединенных Штатах и/или других странах.

Intel, логотип Intel, Intel Inside, логотип Intel Inside, Intel Centrino, логотип Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium и Pentium являются товарными знаками или зарегистрированными товарными знаками компании Intel или ее дочерних компаний в Соединенных Штатах и других странах.

Linux является зарегистрированным товарным знаком Linus Torvalds в Соединенных Штатах и других странах.

Microsoft, Windows, Windows NT и логотип Windows являются товарными знаками корпорации Microsoft в Соединенных Штатах и других странах.

UNIX является зарегистрированным товарным знаком The Open Group в Соединенных Штатах и других странах.

Java и все основанные на Java товарные знаки и логотипы - товарные знаки или зарегистрированные товарные знаки Oracle и/или его филиалов.

Правила и условия для документации продукта

Разрешения для использования этих публикаций предоставляются на следующих условиях.

Применимость

Данные правила и условия являются дополнением к правилам использования для сайта IBM.

Персональное использование

Вы можете воспроизводить эти публикации для персонального некоммерческого использования при условии сохранения всех замечаний о правах собственности. Вам запрещается распространять эти публикации, полностью или по частям, демонстрировать их или создавать из них производные продукты без явного на то согласия от IBM.

Коммерческое использование

Вам предоставляется право воспроизводить эти публикации исключительно в пределах своего предприятия при условии, что будут воспроизведены все замечания об авторских правах. За пределами вашего предприятия вам запрещается распространять эти публикации, полностью или по частям, демонстрировать их или создавать из них производные продукты без явного на то согласия от IBM.

Права

За исключением прав, явным образом предоставляемых настоящим разрешением, никаких иных разрешений, лицензий и прав, ни явных, ни подразумеваемых, в отношении публикаций и любой содержащейся в них информации, данных, программ или иной интеллектуальной собственности, не предоставляется.

IBM оставляет за собой право отозвать разрешения, предоставленные этим документом, если, по мнению IBM, использование публикаций наносит ущерб IBM или, как это установлено IBM, вышеприведенные инструкции не соблюдаются должным образом.

Запрещается загружать, экспортировать или реэкспортировать эту информацию, если при этом не будут полностью соблюдаться все применимые законы и постановления, включая все законы и постановления США, касающиеся экспорта.

IBM НЕ ДАЕТ НИКАКИХ ГАРАНТИЙ ОТНОСИТЕЛЬНО СОДЕРЖАНИЯ ЭТИХ ПУБЛИКАЦИЙ. ПУБЛИКАЦИИ ПРЕДСТАВЛЯЮТСЯ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ (НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ) ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ ОТСУТСТВИЯ НАРУШЕНИЙ, КОММЕРЧЕСКОЙ ПРИГОДНОСТИ ИЛИ СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

Индекс

A

API
setReportHandler 34
альтернативный ввод 29
альтернативный вывод 31
коды ошибок 21
коды состояний 21
количество полей 28
обзор 13, 21
опции выполнения 23, 24
справочник 21
типы полей 28

C

closeImage
обзор процессов API 21
справочник по функциям API 26

E

enumerateParameters
обзор процессов API 21
справочник по функциям API 26

ERM

API 19

execute

обзор процессов API 21
справочник по функциям API 32

G

getErrorDetail
обзор процессов API 21
справочник по функциям API 33

getFieldCount

обзор процессов API 21
справочник по функциям API 28

getFieldTypes

обзор процессов API 21
справочник по функциям API 28

getOption

обзор процессов API 21
справочник по функциям API 23

getParameter

обзор процессов API 21
справочник по функциям API 27

I

IBM SPSS Modeler 1
документация 3
IBM SPSS Modeler Server 1
IBM SPSS Modeler Solution Publisher 7, 8,
19
встраивание в программы 13
обзор 7
initialise
справочник по функциям API 22

initialise (продолжение)

флаги 22
initialise_ext
справочник по функциям API 22
флаги 22
interrupt
обзор процессов API 21
справочник по функциям API 33

O

openImage
обзор процессов API 21
справочник по функциям API 25

P

prepare
обзор процессов API 21
справочник по функциям API 32

S

setAlternativeInput
обзор процессов API 21
справочник по функциям API 29
setAlternativeOutput
обзор процессов API 21
справочник по функциям API 31
setLogFile
обзор процессов API 21
справочник по функциям API 24
setOption
обзор процессов API 21
справочник по функциям API 24
setParameter
обзор процессов API 21
справочник по функциям API 27
setReportHandler
обзор процессов API 21
справочник по функциям API 34

U

UNIX
установка IBM SPSS Modeler Solution
Publisher Runtime 15

Б

библиотека программирования времени
выполнения (CLEMRTL) 13

В

внедрение 7, 19

временная папка

Модуль времени выполнения SPSS
Modeler Solution Publisher 17
встраиваемая предсказательная аналитика
IBM SPSS
API 19
выполнение потоков
использование IBM SPSS Modeler
Solution Publisher 11, 12
выходные узлы 7, 19

Д

деинсталляция
модуль времени выполнения SPSS
Modeler Solution Publisher 18
документация 3

З

закрытие
потоки 26

И

инициализировать
обзор процессов API 21

К

коды ошибок 21
коды состояний 21

Л

локаль
задание с помощью API 22

М

модуль времени выполнения
деинсталляция 18
запуск 17
каталог temp 17
опции 11
установка в UNIX 15
установка в Windows 15
модуль времени выполнения SPSS Modeler
Solution Publisher
деинсталляция 18
запуск 17
системные требования 15
установка 15
Модуль времени выполнения SPSS Modeler
Solution Publisher
каталог temp 17

О

открытие
потоки 25

П

параметры
параметры для выполнения
потоков 12
параметры образа API 26, 27
процедуры обратного вызова API 26
файлы параметров API 25, 26
потоки публикации
IBM SPSS Modeler Solution Publisher 7,
8
примеры
обзор 4
Руководство по прикладным
программам 3
примеры прикладных программ 3

С

сообщения об ошибках
справочник по функциям API 24, 33,
34

У

узел IBM SPSS Modeler Solution
Publisher 19
узел Опубликовать 19

Ф

файл .par 8
файл .pim 8
файлы журналов 24
Функции API
closeImage 21, 26
enumerateParameters 21, 26
execute 21, 32
getErrorDetail 21, 33
getFieldCount 21, 28
getFieldTypes 21, 28
getOption 21, 23
getParameter 21, 27
interrupt 21, 33
openImage 21, 25
prepare 21, 32
setAlternativeInput 21, 29
setAlternativeOutput 21, 31
setLogFile 21, 24
setOption 21, 24
setParameter 21, 27
setReportHandler 21
инициализация 21, 22

Ч

часовой пояс
задание с помощью API 22



Напечатано в Дании