

*IBM SPSS Modeler 17.1 Solution
Publisher*

IBM

Nota

Prima di utilizzare queste informazioni e il relativo prodotto, leggere le informazioni in "Note" a pagina 39.

Informazioni sul prodotto

Questa edizione si applica alla versione 17, release 1, modifica 0 di IBM(r) SPSS(r) Modeler ed a tutte le release e modifiche successive se non diversamente indicato nelle nuove edizioni.

Indice

| | |
|-----------------------------|----------|
| Prefazione | v |
|-----------------------------|----------|

Capitolo 1. Informazioni su IBM SPSS

| | |
|--------------------------|----------|
| Modeler | 1 |
|--------------------------|----------|

| | |
|--|---|
| Prodotti IBM SPSS Modeler | 1 |
| IBM SPSS Modeler | 1 |
| IBM SPSS Modeler Server | 1 |
| IBM SPSS Modeler Administration Console | 2 |
| IBM SPSS Modeler Batch | 2 |
| IBM SPSS Modeler Solution Publisher | 2 |
| Adattatori IBM SPSS Modeler Server per IBM SPSS Collaboration and Deployment Services | 2 |
| Edizioni di IBM SPSS Modeler | 2 |
| Documentazione di IBM SPSS Modeler | 3 |
| Documentazione di SPSS Modeler Professional | 3 |
| Documentazione di SPSS Modeler Premium | 4 |
| Esempi di applicazioni | 5 |
| Cartella Demos | 5 |

Capitolo 2. IBM SPSS Modeler Solution Publisher

| | |
|---|----------|
| Funzionamento di IBM SPSS Modeler Solution Publisher | 7 |
| Pubblicazione di flussi | 8 |

Capitolo 3. Esecuzione di flussi pubblicati

| | |
|--|-----------|
| Utilizzo del file di parametri per personalizzare l'esecuzione del flusso | 12 |
| Integrazione di IBM SPSS Modeler Solution Publisher con altre applicazioni. | 13 |
| Applicazione di esempio IBM SPSS Modeler Solution Publisher | 14 |

Capitolo 4. Installazione di IBM SPSS

Modeler Solution Publisher Runtime

| | |
|--|----|
| Installazione del Runtime. | 15 |
| Requisiti di sistema. | 15 |
| Installazione in ambiente Windows | 15 |
| Installazione in ambiente UNIX. | 16 |
| Risoluzione dei problemi di un'installazione | 17 |
| Avvio del Runtime | 18 |
| Modifica della directory Temp | 18 |
| Disinstallazione di IBM SPSS Modeler Solution Publisher Runtime | 18 |
| Procedura di disinstallazione su computer Windows | 19 |
| Procedura di disinstallazione su computer UNIX | 19 |

Capitolo 5. API di IBM SPSS Embedded Predictive Modeling

Capitolo 6. Riferimenti all'API della libreria di IBM SPSS Modeler Solution

Publisher Runtime.

| | |
|---------------------------------------|-----------|
| Introduzione | 23 |
| Panoramica del processo API | 23 |
| Funzioni API | 24 |

Note

| | |
|-------------------------|-----------|
| Marchi | 40 |
|-------------------------|-----------|

Indice analitico

Prefazione

IBM® SPSS Modeler è l'efficace workbench di data mining aziendale di IBM Corp.. SPSS Modeler consente alle organizzazioni di migliorare le relazioni con i clienti e con il pubblico grazie a un'analisi approfondita dei dati. Le organizzazioni potranno utilizzare le informazioni ottenute tramite SPSS Modeler per mantenere i clienti di valore, cogliere opportunità di vendite incrociate, attrarre nuovi clienti, individuare frodi, diminuire i rischi e migliorare l'offerta di servizi a livello statale.

L'interfaccia visuale di SPSS Modeler favorisce l'applicazione di una competenza di business specifica da parte degli utenti, grazie alla quale sarà possibile ottenere modelli di previsione più efficaci ed una riduzione nei tempi di sviluppo delle soluzioni. SPSS Modeler offre una vasta gamma di tecniche di creazione di modelli, quali previsione, classificazione, segmentazione ed algoritmi per l'individuazione delle associazioni. IBM SPSS Modeler Solution Publisher consente quindi di distribuire a livello aziendale i modelli creati in modo che vengano utilizzati dai responsabili dei processi decisionali oppure inseriti in un database.

Informazioni su IBM Business Analytics

Il software IBM Business Analytics fornisce informazioni complete, coerenti e accurate a cui i responsabili delle decisioni possono affidarsi per ottimizzare le prestazioni di business. Un ampio portafoglio di applicazioni di business intelligence, analisi predittiva, gestione delle prestazioni e delle strategie finanziarie e analisi offre una panoramica chiara, istantanea e interattiva delle prestazioni attuali e la possibilità di prevedere i risultati futuri. Utilizzato in combinazione con potenti soluzioni di settore, prassi consolidate e servizi professionali, questo software consente alle aziende di tutte le dimensioni di ottimizzare la produttività, automatizzare le decisioni senza problemi e fornire risultati migliori.

Come parte di questo portafoglio, il software IBM SPSS Predictive Analytics consente alle aziende di prevedere gli eventi futuri e di agire tempestivamente in modo da migliorare i risultati delle attività di business. Le aziende, gli enti governativi e le università di tutto il mondo si affidano alla tecnologia IBM SPSS perché rappresenta un vantaggio concorrenziale in termini di attrazione, retention ed aumento dei clienti, riducendo al tempo stesso le frodi e limitando i rischi. Incorporando il software IBM SPSS nelle attività quotidiane, le aziende diventano imprese in grado di effettuare previsioni e di gestire ed automatizzare le decisioni, per raggiungere gli obiettivi di business e vantaggi tangibili sulla concorrenza. Per ulteriori informazioni o per contattare un rappresentante, visitare il sito <http://www.ibm.com/spss>.

Supporto tecnico

Il supporto tecnico è a disposizione dei clienti che dispongono di un contratto di manutenzione. I clienti possono contattare il supporto tecnico per richiedere assistenza per l'utilizzo di IBM Corp. o per l'installazione di uno degli ambienti hardware supportati. Per contattare il supporto tecnico, visitare il sito Web IBM Corp. all'indirizzo <http://www.ibm.com/support>. Ricordare che durante la richiesta di assistenza sarà necessario fornire i dati di identificazione personali, i dati relativi alla propria società e il numero del contratto di manutenzione.

Capitolo 1. Informazioni su IBM SPSS Modeler

IBM SPSS Modeler è un insieme di strumenti di data mining che consente di sviluppare rapidamente modelli predittivi con l'ausilio di competenze di business e di eseguirne la distribuzione nelle operazioni di business per migliorare i processi decisionali. Progettato secondo il modello CRISP-DM conforme agli standard di settore, IBM SPSS Modeler supporta l'intero processo di data mining, dai dati a risultati di business migliori.

IBM SPSS Modeler offre numerosi metodi di modellazione ricavati dall'apprendimento automatico, dall'intelligenza artificiale e dalla statistica. I metodi disponibili nella palette Modelli consentono di ricavare nuove informazioni dai dati e di sviluppare modelli predittivi. Ogni metodo ha determinati punti di forza e si presta meglio per particolari tipi di problemi.

SPSS Modeler può essere acquistato come prodotto autonomo oppure utilizzato come client in combinazione con SPSS Modeler Server. È inoltre disponibile una serie di opzioni, come illustrato nelle sezioni seguenti. Per ulteriori informazioni, consultare <http://www.ibm.com/software/analytics/spss/products/modeler/>.

Prodotti IBM SPSS Modeler

La famiglia di prodotti IBM SPSS Modeler e del software associato comprende quanto segue.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- Adattatori IBM SPSS Modeler Server per IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler è una versione del prodotto con funzionalità complete che viene installata ed eseguita sul proprio PC. È possibile eseguire SPSS Modeler in modalità locale come prodotto autonomo oppure in modalità distribuita assieme a IBM SPSS Modeler Server per ottenere una migliore performance su insiemi di dati di grandi dimensioni.

Grazie a SPSS Modeler si possono creare, in modo veloce e intuitivo, modelli predittivi accurati senza ricorrere alla programmazione. La sua avanzata interfaccia visiva permette di visualizzare con facilità il processo di data mining. Grazie alle funzionalità di analisi avanzate incorporate nel prodotto, l'utente potrà rilevare la presenza di pattern e tendenze, che altrimenti rimarrebbero occulti, all'interno dei dati. La modellazione dei risultati e la comprensione dei fattori che li influenzano consente di beneficiare di maggiori opportunità di business e, al contempo, di ridurre i rischi.

SPSS Modeler è disponibile in due edizioni: SPSS Modeler Professional e SPSS Modeler Premium. Per ulteriori informazioni, consultare l'argomento "Edizioni di IBM SPSS Modeler" a pagina 2.

IBM SPSS Modeler Server

SPSS Modeler utilizza un'architettura client/server per distribuire le richieste di operazioni che utilizzano molte risorse a potenti componenti software server, con un conseguente miglioramento della performance su insiemi di dati di grandi dimensioni.

SPSS Modeler Server è un prodotto con licenza separata che viene eseguito continuamente in modalità di analisi distribuita su un host server insieme a una o più installazioni IBM SPSS Modeler. Una configurazione di questo tipo consente a SPSS Modeler Server di ottenere prestazioni migliori quando si lavora su insiemi di dati di grandi dimensioni, in quanto le operazioni che richiedono un utilizzo consistente della memoria possono essere eseguite sul server senza scaricare i dati sul computer client. IBM SPSS Modeler Server offre inoltre il supporto delle funzionalità di ottimizzazione SQL e di modellazione nel database, garantendo ulteriori benefici dal punto di vista delle prestazioni e del livello di automazione.

IBM SPSS Modeler Administration Console

Modeler Administration Console è un'applicazione grafica per la gestione di molte delle opzioni di configurazione di SPSS Modeler Server, la cui configurazione può avvenire, inoltre, mediante un file delle opzioni. L'applicazione fornisce un'interfaccia utente di console per monitorare e configurare le installazioni di SPSS Modeler Server ed è disponibile gratuitamente per i clienti esistenti di SPSS Modeler Server. L'applicazione può essere installata solo sui computer Windows; tuttavia, può gestire un server installato su qualsiasi piattaforma supportata.

IBM SPSS Modeler Batch

Nonostante il data mining sia generalmente un processo di tipo interattivo, è possibile eseguire SPSS Modeler da una riga di comando senza il bisogno di ricorrere all'interfaccia utente grafica. Poniamo, ad esempio, che si debbano svolgere varie attività laboriose e ripetitive che non richiedono l'intervento di un utente. SPSS Modeler Batch è una versione speciale del prodotto che supporta l'intera gamma di funzionalità analitiche di SPSS Modeler senza richiedere l'accesso all'interfaccia utente normale. Per utilizzare SPSS Modeler Batch è necessario installare SPSS Modeler Server.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher è uno strumento che consente di creare una versione a pacchetto di un flusso SPSS Modeler che potrà essere eseguito da un motore di runtime esterno oppure incorporato in una applicazione esterna. Questo permette di pubblicare e sottoporre a deployment stream SPSS Modeler completi in ambienti in cui SPSS Modeler non è installato. SPSS Modeler Solution Publisher è distribuito come parte del servizio IBM SPSS Collaboration and Deployment Services - Scoring, per cui è necessario procurarsi una licenza separata. Insieme alla licenza, si riceve SPSS Modeler Solution Publisher Runtime, che consente di eseguire i flussi pubblicati.

Per ulteriori informazioni su SPSS Modeler Solution Publisher, consultare la documentazione IBM SPSS Collaboration and Deployment Services. IBM SPSS Collaboration and Deployment Services Knowledge Center contiene le sezioni denominate "IBM SPSS Modeler Solution Publisher" e "IBM SPSS Analytics Toolkit."

Adattatori IBM SPSS Modeler Server per IBM SPSS Collaboration and Deployment Services

È disponibile una serie di adattatori per IBM SPSS Collaboration and Deployment Services che abilitano l'interazione di SPSS Modeler e SPSS Modeler Server con un repository IBM SPSS Collaboration and Deployment Services. In questo modo, un flusso SPSS Modeler sottoposto a deployment sul repository potrà essere condiviso da più utenti oppure risulterà accessibile dall'applicazione thin client IBM SPSS Modeler Advantage. L'adattatore va installato sul sistema che ospita il repository.

Edizioni di IBM SPSS Modeler

SPSS Modeler è disponibile nelle edizioni seguenti.

SPSS Modeler Professional

SPSS Modeler Professional contiene tutti gli strumenti necessari per utilizzare la maggior parte dei tipi di dati strutturati, quali comportamenti e interazioni registrati in sistemi CRM, dati demografici, dati sulle vendite e sul comportamento d'acquisto.

SPSS Modeler Premium

SPSS Modeler Premium è un prodotto con licenza separata che amplia l'ambito di utilizzo di SPSS Modeler Professional aggiungendo il supporto di dati speciali, quali quelli usati per l'analisi delle entità o dei social network, e di dati di testo non strutturati. SPSS Modeler Premium comprende i seguenti componenti.

IBM SPSS Modeler Entity Analytics aggiunge una dimensione supplementare alle analisi predittive di IBM SPSS Modeler. Se l'analisi predittiva tenta di prevedere il comportamento futuro sulla base di dati precedenti, l'analisi dell'entità si concentra sul miglioramento della coerenza dei dati correnti risolvendo i conflitti tra gli stessi record. Un'identità può essere di un individuo, un'organizzazione, un oggetto o qualsiasi altra entità per cui possa esistere ambiguità. La risoluzione dell'identità può essere essenziale in diversi campi, tra cui la gestione delle relazioni con i clienti, il rilevamento di frodi, il riciclaggio di denaro e la sicurezza nazionale e internazionale.

IBM SPSS Modeler Social Network Analysis trasforma le informazioni sulle relazioni in campi che caratterizzano il comportamento sociale di individui e gruppi. Facendo leva sui dati che descrivono le relazioni esistenti nelle reti sociali, IBM SPSS Modeler Social Network Analysis riesce a individuare i leader in grado di influenzare il comportamento degli altri membri della rete. Consente inoltre di stabilire quali individui della rete sono maggiormente influenzati dagli altri membri. La combinazione di questi risultati ad altre misurazioni permette di delineare profili complessi degli individui su cui basare dei modelli predittivi. I modelli che contengono informazioni sociali generano risultati più accurati rispetto agli altri.

IBM SPSS Modeler Text Analytics utilizza tecnologie linguistiche avanzate e di NLP (Natural Language Processing) per elaborare rapidamente una grande varietà di dati di testo non strutturati, estrarre ed organizzare i concetti chiave e raggruppare tali concetti in categorie. È quindi possibile combinare i concetti e le categorie estratti con dati strutturati esistenti, per esempio dati demografici, e applicarli alla modellazione utilizzando la suite completa degli strumenti di data mining di IBM SPSS Modeler per prendere decisioni migliori e più mirate.

Documentazione di IBM SPSS Modeler

La documentazione nel formato guida in linea è disponibile nel menu Aiuto di SPSS Modeler. Sono incluse la documentazione per SPSS Modeler, SPSS Modeler Server, nonché la Guida alle applicazioni (nota anche come Esercitazione) e altro materiale di supporto.

La documentazione completa in formato PDF dei singoli prodotti, istruzioni di installazione comprese, è disponibile nella cartella *Documentation* del DVD di ciascun prodotto. I documenti per l'installazione possono essere anche scaricati dal Web, all'indirizzo <http://www.ibm.com/support/docview.wss?uid=swg27043831>.

La documentazione in entrambi i formati è anche disponibile nel Knowledge Center SPSS Modeler all'indirizzo http://www-01.ibm.com/support/knowledgecenter/SS3RA7_17.0.0.0.

Documentazione di SPSS Modeler Professional

La documentazione completa di SPSS Modeler Professional, escluse le istruzioni di installazione, è la seguente.

- **IBM SPSS Modeler - Guida per l'utente.** Introduzione generale all'utilizzo di SPSS Modeler che illustra come creare flussi di dati, gestire valori mancanti, generare espressioni CLEM, utilizzare progetti e report e assemblare flussi per la distribuzione tramite IBM SPSS Collaboration and Deployment Services, le applicazioni predittive o IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler - Nodi origine, elaborazione e output.** Descrizioni di tutti i nodi utilizzati per leggere, elaborare e generare dati di output in vari formati, ovvero di nodi ad eccezione dei nodi Modelli.
- **IBM SPSS Modeler - Nodi di modellazione.** Descrizioni di tutti i nodi utilizzati per creare modelli data mining. IBM SPSS Modeler offre numerosi metodi di modellazione ricavati dall'apprendimento automatico, dall'intelligenza artificiale e dalla statistica.
- **IBM SPSS Modeler Algorithms Guide.** Descrizione dei fondamenti di matematica per i metodi di modellazione utilizzati in IBM SPSS Modeler. Questa guida è disponibile solo in formato PDF.
- **IBM SPSS Modeler Applications Guide.** Gli esempi inclusi in questa guida forniscono indicazioni mirate e sintetiche su specifici metodi e tecniche di modellazione. Una versione in linea di questa guida è inoltre disponibile dal menu Aiuto. Per ulteriori informazioni, consultare l'argomento "Esempi di applicazioni" a pagina 5.
- **IBM SPSS Modeler Python - Script e automazione.** Informazioni sulle modalità di automazione del sistema tramite script Python, incluse le proprietà che è possibile utilizzare per manipolare nodi e flussi.
- **IBM SPSS Modeler - Guida alla distribuzione.** Informazioni sull'esecuzione di flussi e scenari IBM SPSS Modeler come fasi dell'elaborazione di lavori in IBM SPSS Collaboration and Deployment Services Deployment Manager.
- **IBM SPSS Modeler - Guida per lo sviluppatore CLEF.** CLEF consente di integrare programmi di terze parti come routine di elaborazione dei dati o algoritmi di modellazione come nodi in IBM SPSS Modeler.
- **IBM SPSS Modeler - Guida al mining nel database.** Informazioni sulle modalità per utilizzare al meglio la potenza del database in uso al fine di ottenere prestazioni migliori ed estendere la gamma di funzionalità analitiche tramite algoritmi di terze parti.
- **IBM SPSS Modeler Server - Guida della performance e amministrazione.** Informazioni su come configurare e amministrare IBM SPSS Modeler Server.
- **IBM SPSS Modeler Administration Console - Guida per l'utente.** Informazioni sull'installazione e l'utilizzo dell'interfaccia utente della console per il monitoraggio e la configurazione di IBM SPSS Modeler Server. La console viene implementata come plug-in dell'applicazione Deployment Manager.
- **IBM SPSS Modeler - Guida CRISP-DM.** Guida passo a passo al data mining tramite la metodologia CRISP-DM con SPSS Modeler.
- **IBM SPSS Modeler Batch - Guida per l'utente.** Guida completa all'utilizzo di IBM SPSS Modeler in modalità batch, contenente dettagli per l'esecuzione della modalità batch e gli argomenti della riga di comando. Questa guida è disponibile solo in formato PDF.

Documentazione di SPSS Modeler Premium

La documentazione completa di SPSS Modeler Premium, escluse le istruzioni di installazione, è la seguente.

- **IBM SPSS Modeler Entity Analytics User Guide.** Contiene informazioni per l'utilizzo dell'analisi delle entità con SPSS Modeler; descrive l'installazione e la configurazione di repository, i nodi Entity Analytics e le attività amministrative.
- **IBM SPSS Modeler Social Network Analysis User Guide.** Guida che spiega come eseguire l'analisi dei social network con SPSS Modeler; comprende l'analisi di gruppo e l'analisi di diffusione.
- **SPSS Modeler Text Analytics - Guida per l'utente.** Contiene informazioni per l'utilizzo di analisi di testo con SPSS Modeler; descrive i nodi di text mining, il workbench interattivo, i modelli e altre risorse.

Esempi di applicazioni

Mentre gli strumenti per il data mining di SPSS Modeler consentono di risolvere un'ampia gamma di problemi a livello di business e organizzativo, gli esempi di applicazioni forniscono indicazioni mirate e sintetiche su specifici metodi e tecniche di modellazione. Gli insiemi di dati utilizzati negli esempi hanno dimensioni molto più limitate rispetto agli enormi archivi di dati gestiti da alcuni data miner, ma i concetti e i metodi coinvolti sono rapportabili alle applicazioni del mondo reale.

È possibile accedere agli esempi facendo clic su **Esempi di applicazioni** nel menu Aiuto di SPSS Modeler. I file di dati e i flussi di esempio sono installati nella cartella *Demos* nella directory di installazione del prodotto. Per ulteriori informazioni, consultare l'argomento "Cartella Demos".

Esempi di modellazione del database. Vedere gli esempi nella *IBM SPSS Modeler Guida al mining nel database*.

Esempi di script. Vedere gli esempi nella *IBM SPSS Modeler Guida per script e automazione*.

Cartella Demos

I file di dati e i flussi di esempio utilizzati negli esempi di applicazioni sono installati nella cartella *Demos* nella directory di installazione del prodotto. È possibile accedere a questa cartella anche dal gruppo di programmi IBM SPSS Modeler nel menu Start di Windows oppure facendo clic su *Demos* nell'elenco delle directory recenti nella finestra di dialogo Apri file.

Capitolo 2. IBM SPSS Modeler Solution Publisher

IBM SPSS Modeler Solution Publisher è un potente strumento che consente di integrare i risultati di data mining nel processo di business per risolvere i problemi del mondo reale. Con IBM SPSS Modeler Solution Publisher è possibile creare una versione "pacchetto" del flusso che può essere eseguita da un motore di runtime esterno o incorporata in un'applicazione esterna. Questo consente di eseguire la distribuzione dei flussi di modellazione dei dati in un ambiente di produzione, per supportare i processi di business e fornire ai responsabili dei processi decisionali le conoscenze acquisite con il mining dei dati.

Anziché limitarsi ad esportare file di modello (come PMML), IBM SPSS Modeler Solution Publisher permette di pubblicare e distribuire *flussi completi di IBM SPSS Modeler*. Ciò significa che è possibile effettuare la preparazione dei dati ed eseguire operazioni su record e campi, per esempio aggregare dati, selezionare record o derivare nuovi campi, prima di creare previsioni basate su un modello. Prima di salvare i dati, è quindi possibile elaborare ulteriormente i risultati del modello, eseguendo semplicemente il flusso pubblicato.

Nota: la pubblicazione avviene mediante la scheda Pubblica del nodo di esportazione appropriato. Per ulteriori informazioni consultare l'argomento "Pubblicazione di flussi" a pagina 8. I flussi salvati in versioni del prodotto precedenti alla 11.0 utilizzavano un nodo Publisher separato. Il nodo è ancora supportato per la compatibilità con le versioni precedenti, ma non è più disponibile nella palette dei nodi.

Licenza di IBM SPSS Modeler Solution Publisher

Nota: IBM SPSS Modeler Solution Publisher è distribuito come parte integrante di IBM SPSS Collaboration and Deployment Services Scoring Service, per cui è necessaria una licenza separata. <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Quando si acquista una licenza, separatamente viene fornito IBM SPSS Modeler Solution Publisher Runtime che consente di eseguire flussi pubblicati. Per ulteriori informazioni consultare l'argomento "Installazione del Runtime" a pagina 15.

Funzionamento di IBM SPSS Modeler Solution Publisher

La distribuzione di una soluzione tramite IBM SPSS Modeler Solution Publisher implica due fasi: la *pubblicazione* di un flusso e l'*esecuzione* di un flusso.

Pubblicazione. Il processo di data mining porta infine a un modello che fornisce una buona soluzione del problema di business. Sarà possibile utilizzare tale modello ed applicarlo al processo di business specifico. Quando si pubblica un flusso, una descrizione dettagliata del flusso viene scritta sul disco (come file *immagine* e file di *parametri*). Per ulteriori informazioni consultare l'argomento "Pubblicazione di flussi" a pagina 8.

Esecuzione. Dopo la pubblicazione del flusso, è possibile ricreare il processo implementato nel flusso eseguendo il flusso pubblicato. Questa operazione si esegue utilizzando IBM SPSS Modeler Runtime autonomo (*modelerrun.exe*) oppure sviluppando un'applicazione che utilizza la libreria di IBM SPSS Modeler Runtime per eseguire il flusso. Per eseguire flussi al di fuori di IBM SPSS Modeler (tramite il Runtime o un'applicazione personalizzata), è necessario prima installare IBM SPSS Modeler Solution Publisher Runtime. Per ulteriori informazioni vedere l'argomento "Installazione del Runtime" a pagina 15.

Publicazione di flussi

La pubblicazione dei flussi viene eseguita direttamente da IBM SPSS Modeler utilizzando uno dei nodi di esportazione standard: Database, File flat, Esporta Statistics, Esporta IBM SPSS Data Collection , Esporta SAS, Excel e i nodi Esporta XML . Il tipo di nodo di esportazione determina il formato dei risultati da scrivere ogni volta che il flusso pubblicato viene eseguito utilizzando IBM SPSS Modeler Solution Publisher Runtime o un'applicazione esterna. Per esempio, se si desidera scrivere i risultati in un database ogni volta che si esegue il flusso pubblicato, utilizzare un nodo di esportazione del database.

Per pubblicare un Flusso

1. Aprire o creare un flusso come di consueto, quindi collegare un nodo di esportazione alla fine.
2. Nella scheda Pubblica nodo di esportazione, specificare un nome di base per i file pubblicati (ovvero il nome file a cui saranno aggiunte le estensioni *.pim*, *.par* and *.xml*).
3. Fare clic sul pulsante **Pubblica** per pubblicare il flusso oppure selezionare **Pubblica flusso per pubblicare automaticamente** il flusso ogni volta che il nodo viene eseguito.

Nome pubblicato. Specificare il nome di base per i file immagine e dei parametri pubblicati.

- Il **file immagine** (**.pim*) contiene tutte le informazioni necessarie affinché Runtime possa eseguire il flusso pubblicato esattamente come era al momento dell'esportazione. Se non è necessario modificare le impostazioni del flusso quali la sorgente dati di input o il file dati di output, è possibile eseguire la distribuzione solo del file immagine.
- The **file di parametri** (**.par*) contiene informazioni configurabili relative a sorgenti dati, file output e opzioni di esecuzione. Se si desidera controllare l'input o l'output del flusso senza pubblicare nuovamente il flusso, sarà necessario il file di parametri così come il file immagine.
- Il **file di metadati** (**.xml*) descrive gli input e gli output dell'immagine e dei rispettivi modelli di dati. Viene utilizzato da applicazioni che incorporano la libreria runtime e che devono conoscere la struttura dei dati di input e output.

Nota: questo file viene generato solo se si seleziona la casella **Pubblica metadati**.

Pubblica parametri. Se necessario, è possibile includere parametri di flusso nel file **.par*. È possibile modificare questi valori di parametri di flusso modificando il file **.par* quando si esegue l'immagine, oppure tramite l'API runtime.

Questa casella di controllo attiva il pulsante **Parametri**. Facendo clic sul pulsante viene visualizzata la finestra di dialogo Pubblica parametri.

Scegliere i parametri da includere nell'immagine pubblicata selezionando la rispettiva casella di controllo nella colonna **Pubblica**.

Durante esecuzione del flusso. Specifica se il flusso viene pubblicato automaticamente quando viene eseguito il nodo.

- **Esporta dati.** Eseguire il nodo di esportazione come di consueto, senza pubblicare il flusso (di fatto il nodo viene eseguito in IBM SPSS Modeler come se IBM SPSS Modeler Solution Publisher non fosse disponibile). Se si seleziona questa opzione, il flusso non viene pubblicato finché non lo si pubblica esplicitamente facendo clic sul pulsante **Pubblica** nella finestra di dialogo del nodo di esportazione. In alternativa, è possibile pubblicare il flusso corrente utilizzando lo strumento Pubblica disponibile sulla barra degli strumenti oppure utilizzando uno script.
- **Pubblica flusso.** Pubblica il flusso per la distribuzione utilizzando IBM SPSS Modeler Solution Publisher. Selezionare questa opzione se si desidera pubblicare automaticamente il flusso ogni volta che viene eseguito.

Commenti

- Se si intende eseguire il flusso pubblicato con dati nuovi o aggiornati, è importante tenere presente che l'ordine dei campi nel file di input deve essere uguale all'ordine dei campi nel file di input del nodo origine specificato nel flusso pubblicato.
- In caso di pubblicazione in applicazioni esterne, è necessario considerare la ridenominazione dei campi o il filtro di campi estranei per garantire la conformità ai requisiti di input. È possibile eseguire entrambe le operazioni utilizzando un nodo Filtro prima del nodo di esportazione.

Capitolo 3. Esecuzione di flussi pubblicati

Il principale impatto dell'esecuzione di un flusso pubblicato è sempre quella di generare un insieme di dati. I dati possono essere memorizzati su disco o come database. I flussi pubblicati non possono generare grafici, tabelle, modelli generati oppure altri tipi di output senza dati.

È possibile eseguire i flussi pubblicati con IBM SPSS Modeler Solution Publisher utilizzando il programma IBM SPSS Modeler Solution Publisher Runtime. Il programma di runtime, *modelerrun.exe*, viene avviato dalla riga di comando con le opzioni che indicano il file immagine e facoltativamente il file di parametri da eseguire. Il comando di runtime è il seguente:

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

dove le voci di comando sono illustrate nella tabella seguente.

Tabella 1. Voci di comando runtime

| Voci di comando | Descrizione |
|------------------|---|
| -nobanner | elimina il messaggio di testo di avvio del runtime (facoltativo). |
| <parameter-file> | è il nome del file di parametri (facoltativo). |
| <options> | è un elenco separato da virgole di coppie option=value che specificano le opzioni di esecuzione per il runtime. Le opzioni vengono dettagliate di seguito (facoltativo). Nota: Dopo la virgola non vi devono essere spazi vuoti. |
| <image-file> | è il nome del file immagine da eseguire (obbligatorio). |

Le opzioni di esecuzione disponibili sono illustrate nella tabella seguente.

Tabella 2. Opzioni di runtime valide.

| Nome | Valore predefinito | Descrizione |
|-----------------------|--------------------|--|
| max_file_size | -1 | Dimensione massima dei file salvati. Un valore pari a -1 indica nessun limite. |
| max_sql_string_length | 2,048 | Lunghezza massima di una stringa importata dal database con SQL. I valori stringa superiori vengono troncati a destra senza avvertire l'utente. L'intervallo valido è compreso tra 1 e 65.535 caratteri. |
| memory_usage | 100 | Moltiplicatore per allocazione di memoria dinamica. Regolare questo valore diminuendolo o aumentandolo per regolare l'utilizzo della memoria totale del server. |
| temp_directory | " | Specifica la directory da utilizzare per i file temporanei. |
| request_passwords | Y | Controlla se viene chiesta una password per il database quando viene eseguito un flusso pubblicato che richiede l'accesso al database (Y/N). |

Utilizzo del file di parametri per personalizzare l'esecuzione del flusso

Quando un flusso viene pubblicato, il file immagine contiene informazioni sulla sorgente dati e sulla destinazione dei dati di output che sono stati selezionati nel flusso originale. Tuttavia, spesso conviene eseguire un flusso rispetto a dati provenienti da una sorgente dati diversa oppure reindirizzare i dati di output su una destinazione diversa. È possibile eseguire questa operazione modificando il file di parametri del flusso.

Si tratta di un'operazione molto importante quando il flusso pubblicato viene eseguito su un computer diverso da quello in cui era stato creato. Sarà necessario aggiornare le posizioni dei file di input e di output per riflettere la struttura del computer di destinazione. Tenere presente che i flussi pubblicati non supportano i relativi percorsi, ed è per questo che è molto importante verificare le posizioni dei dati nel file di parametri. Se si specificano i percorsi relativi nelle opzioni del nodo Esporta quando si pubblica il flusso, IBM SPSS Modeler li converte in percorsi assoluti prima di creare i file immagine e di parametri per il flusso pubblicato.

È inoltre possibile utilizzare il file di parametri per impostare le opzioni di esecuzione e modificare i valori dei parametri dei flussi.

I valori dei parametri vengono specificati nel file di parametri come coppie `<name.attribute>=<value>`. In alcuni, il nome del parametro contiene un ID digitale per distinguere i nodi dello stesso tipo nel flusso. Per esempio, in un flusso in cui input e output fanno riferimento alla connessione di un database, i parametri che descrivono la connessione di input possono essere `dbconn0.datasource`, `dbconn0.user`, e così via e i parametri associati alla connessione di output possono essere `dbconn1.datasource`, `dbconn1.user`, ecc. I valori sono sempre racchiusi tra virgolette. Il file di parametri contiene i seguenti parametri, che possono essere modificati per personalizzare l'esecuzione del flusso.

Tabella 3. Parametri di esecuzione.

| Parametro | Scopo |
|---|--|
| <code>dbconnx.datasource</code> | Nome sorgente dati, Data source name (DSN). |
| <code>dbconnx.user</code> | Nome utente per database con accesso limitato. |
| <code>dbconnx.password</code> | Password per database con accesso limitato |
| <code>dbconnx.epassword</code> | Password codificata per database con accesso limitato. Per generare una password codificata, selezionare Codifica password dal menu Strumenti dell'interfaccia utente di IBM SPSS Modeler. Copiare ed incollare la password codificata se necessario. |
| <code>dbconnx.password_required</code> | Flag che indica se richiedere una password per questa connessione al database (Il valore è "S" o "N".) Impostato automaticamente su "N" se non viene specificata una password al momento della pubblicazione. |
| <code>dbconnx.catalog</code> | Catalogo associato alla connessione al database. |
| <code>dbtablex.name</code> | Nome della tabella di base. <i>Nota:</i> l'ordine dei campi in una nuova tabella deve essere uguale all'ordine dei campi nel flusso pubblicato in origine. |
| <code>dbtablex.schema</code> | Nome dello schema. |
| <code>dbtablex.catalog</code> | Catalogo associato con la tabella corrente. |
| <code>filex.name</code> | Nome del file per il file di dati. <i>Nota:</i> l'ordine dei campi in un nuovo file di input deve essere uguale all'ordine dei campi nel flusso pubblicato in origine. |
| <code>filex.path</code> | Posizione del file (nome della directory). |
| <code>filex.field_names_included</code> | Per i file in formato testo, il flag che indica se i nomi dei campi sono inclusi come prima riga del file. (Il valore è "S" o "N".) |

Tabella 3. Parametri di esecuzione (Continua).

| Parametro | Scopo |
|---|---|
| <code>filex.field_separator</code> | Per i file in formato testo, i caratteri utilizzati per separare i valori di campo. |
| <code>filex.decimal_separator</code> | Per i file in formato testo, il carattere utilizzato per il punto decimale. |
| <code>options.angle_in_radians</code> | Flag che indica se i radianti vengono utilizzati come unità di misura nelle espressioni trigonometriche CLEM. (Il valore è "S" o "N".) |
| <code>options.date_2digit_baseline</code> | Definisce il secolo delle date con l'anno a due cifre. |
| <code>options.date_baseline</code> | Anno di riferimento (sempre 1 gen) utilizzato dalle funzioni di data CLEM che operano con una singola data. |
| <code>options.time_rollover</code> | Flag che indica se le differenze di tempo negative si riferiscono al passato. |
| <code>options.decimal_separator</code> | Il carattere a punto decimale predefinito nei file di testo. |
| <code>options.time_format</code> | Formato di ora utilizzato quando le stringhe vengono interpretate come orari dalle funzioni di ora di CLEM. |
| <code>options.date_format</code> | Formato di data utilizzato quando le stringhe vengono interpretate come date dalle funzioni di data di CLEM. |
| <code>options.timestamp_format</code> | Formato utilizzato durante la lettura dei campi Timestamp come stringhe da sorgenti di dati ODBC. |
| <code>paramx.value</code> | Valore del parametro del flusso. <i>Nota:</i> il nome del parametro, la descrizione ed il tipo di archiviazione sono inclusi per riferimento ma non possono essere cambiati. Il valore deve essere valido per il tipo. |

Integrazione di IBM SPSS Modeler Solution Publisher con altre applicazioni

Oltre al motore di runtime autonomo, IBM SPSS Modeler Solution Publisher dispone di una libreria di programmazione del runtime (CLEMRTL) che consente ad altri programmi di controllare l'esecuzione del flusso di IBM SPSS Modeler Solution Publisher. È possibile chiamare le procedure CLEMRTL nei programmi client scritti in C e C++. Per utilizzare CLEMRTL, è necessario includere il file di intestazione `clemrtl.h` (disponibile nella cartella `..\installation\clemrtl\include`) e accertarsi di collegare il file della libreria appropriato alla piattaforma di sviluppo nell'applicazione quando viene creato. Il file `clemrtl.dll` è disponibile nella cartella `..\installation\bin`.

Qualsiasi file di origine che fa riferimento a procedure della libreria deve includere l'intestazione `clemrtl.h`. Questo file di intestazione contiene prototipi ANSI C per le procedure della libreria e consente di definire macro utili. Non richiede l'inclusione di nessun'altra intestazione oltre a quelle richieste dal programma. Per evitare sovrapposizioni di nomi, tutti i tipi di libreria e i nomi di funzioni iniziano con `clemrtl_` e tutti i nomi di macro hanno il prefisso `CLEMRTL_`.

CLEMRTL genera rapporti che contengono informazioni utili che devono essere ritrasmesse in qualche modo all'applicazione. CLEMRTL dispone di tre meccanismi per gestire questo tipo di messaggi:

- I messaggi possono essere scritti in un file di log specificato dalla funzione `clemrtl_setLogFile()`.
- L'applicazione può recuperare i dettagli relativi all'ultimo errore tramite la funzione `clemrtl_getErrorDetail()`.
- L'applicazione dispone della propria procedura di gestione dei report tramite la funzione `clemrtl_setReportHandler()`.

Il sistema di runtime contiene un catalogo di messaggi localizzati per i report. Qualsiasi report trasmesso all'applicazione include la stringa del messaggio localizzato. I report includono anche il codice del report

in modo che l'applicazione possa scegliere se interpretare e presentare il messaggio diversamente. Per ottenere messaggi dettagliati localizzati, l'applicazione deve inviare il file appropriato *messages.cfg* in una directory *config*. È possibile trovare diverse versioni di *messages.cfg* per opzioni internazionali diverse nella directory `<installdir>/config/<locale>`, dove `<installdir>` è la directory in cui è installato IBM SPSS Modeler Solution Publisher e `<locale>` sono le opzioni internazionali desiderate. Selezionare la versione appropriata di *messages.cfg* ed aggiungerla a una sottodirectory *config* nella directory del proprio progetto.

Il sistema di runtime non necessita di un ambiente o di impostazioni di registro speciali per operare correttamente. Le librerie dipendenti devono essere distribuite con un'applicazione collegata a CLEMRTL. Questi file sono inclusi nella directory `<installdir>/bin`.

Requisiti

La tabella seguente riporta le piattaforme ed i compilatori sottoposti a test tramite la libreria di programmazione runtime.

Tabella 4. Piattaforme e compilatori.

| Piattaforma | Compilatori consigliato |
|------------------|--|
| AIX | IBM XL C/C++ Enterprise Edition V8.0 per AIX |
| Solaris a 64 bit | Sun Studio 11 5.8 Patch 121017-06 |
| Linux a 64 bit | gcc 3.2.3, compilatore Intel C++ 9.1.043 |
| Windows | Visual Studio 2005 |

Applicazione di esempio IBM SPSS Modeler Solution Publisher

Un'applicazione di esempio, *dlltest.c*, è inclusa nella cartella `\clemtr\demo` della directory di installazione di IBM SPSS Modeler Solution Publisher. Utilizzare l'applicazione di esempio per familiarizzare con l'integrazione della libreria di runtime con altre applicazioni.

Le informazioni utilizzate dall'applicazione di esempio sono incluse nella cartella `\clemtr\demo`; per le istruzioni, vedere il file *README.txt*. La compilazione di un'applicazione varia in base alla piattaforma, quindi vengono inclusi anche makefile di esempio.

Capitolo 4. Installazione di IBM SPSS Modeler Solution Publisher Runtime

Installazione del Runtime

L'installazione di IBM SPSS Modeler Solution Publisher Runtime consente di eseguire i flussi pubblicati. I flussi pubblicati non dipendono dal sistema operativo in uso. È pertanto possibile eseguire qualsiasi flusso pubblicato su qualsiasi piattaforma supportata dal Runtime.

La funzionalità che consente di pubblicare flussi da IBM SPSS Modeler viene installata con IBM SPSS Modeler. Si noti che i file pubblicati devono essere riesportati con ciascuna release di IBM SPSS Modeler. Ad esempio, un file esportato utilizzando una release precedente non funzionerà con la versione 17 di IBM SPSS Modeler Solution Publisher Runtime.

Requisiti di sistema

Per l'installazione di IBM SPSS Modeler Solution Publisher Runtime sono necessari i seguenti requisiti:

- **Sistema operativo.** Windows Server 2008 Standard Edition per sistemi x86 a 32-bit; Windows Server 2008 Enterprise Edition per sistemi x64 a 64-bit; Windows Server 2008 Enterprise Edition R2 per sistemi x64 a 64-bit; Windows Server 2003 Standard Edition R2 per sistemi x86 a 32-bit o sistemi x64 a 64-bit; IBM AIX® 6.1 o 7.1 per sistemi POWER a 64-bit; Oracle Solaris™ 9.x o 10 per sistemi SPARC a 64-bit; Red Hat Enterprise Linux 5.x per sistemi x86 a 32-bit; Red Hat Enterprise Linux 5.x o 6.x per sistemi x64 a 64-bit o sistemi IBM System z; Red Hat Enterprise Linux Advanced Platform 5.x per sistemi x86 a 32-bit; Red Hat Enterprise Linux Advanced Platform 5.x o 6.x per sistemi x64 a 64-bit o sistemi IBM System z; SuSE Linux Enterprise Server 10 o 11 per sistemi x64 a 64-bit o sistemi IBM System z
- **Hardware.** Processore Pentium® di categoria equivalente o superiore (per Windows a 32-bit); famiglia di processori x64 (AMD 64 e EM64T; per Windows a 64-bit Windows), con velocità di 1GHz o superiore Processore PowerPC, con velocità di 233MHz o superiore e IBM System p per IBM AIX; processore Itanium per HP-UX; UltraSPARC II (o superiore) per Solaris; processore Pentium o di categoria equivalente o superiore per Linux a 32-bit; famiglia di processori x64 (AMD 64 e EM64T) o IBM s390x System z per Linux a 64-bit
- **Spazio minimo libero su disco.** 2 gigabyte (GB) consigliati.
- **RAM.** 4 GB consigliati.
- **Ambiente virtuale.** Gli ambienti virtuali seguenti supportano IBM SPSS Modeler Server.
 - IBM PowerVM Hypervisor per IBM AIX
 - IBM PR/SM o z/VM per Linux per IBM System z
 - VMWare ESX Server 4.1
 - VMWare vSphere 4.0
- **Ambiente cluster.** Gli ambienti cluster seguenti supportano IBM SPSS Modeler Server.
 - IBM HACMP/RSCT per IBM AIX
 - IBM WebSphere App Server a 64 bit
 - Microsoft Cluster Service per Windows
 - Oracle Solaris Cluster
 - Oracle WebLogic App Server a 64 bit
 - Red Hat Cluster Suite per Red Hat Enterprise Linux

Installazione in ambiente Windows

Nella seguente sezione è descritta la procedura per installare SPSS Modeler Solution Publisher Runtime sul sistema operativo Windows.

1. Inserire il disco di installazione IBM SPSS Modeler Solution Publisher nell'unità, oppure accedere al supporto di installazione con altri strumenti.
2. Nel menu di installazione, scegliere **Installa IBM SPSS Modeler Solution Publisher Runtime** o **Installa IBM SPSS Modeler Solution Publisher Runtime for Win64**, a seconda del sistema su cui viene eseguita l'installazione.
3. Nella procedura guidata di installazione, fare clic su **Avanti** per iniziare.
4. Seguire le istruzioni visualizzate sullo schermo. Per continuare, fare clic su **Avanti**.
5. Dopo aver specificato tutte le opzioni, è possibile procedere all'installazione. Fare clic su **Installa** per iniziare a trasferire i file.
6. Dopo che tutti i file sono stati installati, fare clic su **Fine**.

Installazione in ambiente UNIX

Nella seguente sezione sono descritti i requisiti di sistema aggiuntivi e la procedura per installare IBM SPSS Modeler Solution Publisher Runtime sul sistema operativo UNIX.

Requisiti aggiuntivi per UNIX

È necessario verificare che i limiti del kernel sul sistema siano sufficienti a consentire il funzionamento di IBM SPSS Modeler Server. I parametri `ulimit` relativi a dati, memoria e file sono particolarmente importanti e dovrebbero essere impostati su un valore illimitato nell'ambiente IBM SPSS Modeler Server. A questo scopo, eseguire le operazioni seguenti:

1. Aggiungere i seguenti comandi a `modelersrv.sh`:

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

Inoltre, impostare il limite stack sul valore massimo consentito dal proprio sistema (`ulimit -s XXXX`), ad esempio:

```
ulimit -s 64000
```

2. Riavviare IBM SPSS Modeler Server.

Per permettere al programma di installazione di decomprimere i file di installazione è necessario che le utility `gzip` e `GNU cpio` siano installate che nel PATH. Inoltre, sulle macchine con SPSS Modeler Server, è necessario impostare la locale su `EN_US.UTF-8`.

Procedura di installazione su computer UNIX

Le seguenti istruzioni di installazione si riferiscono alla versione IBM SPSS Modeler Solution Publisher Runtime per UNIX. È necessario disporre di autorizzazioni in lettura e scrittura per la directory di installazione di destinazione. Accedere pertanto con un account che disponga di autorizzazioni sufficienti.

Nota: è necessario essere collegati come `root` per eseguire l'installazione.

1. Inserire il disco di installazione nel lettore, assicurandosi di poter accedere a questa unità. Su Solaris deve essere in esecuzione il gestore dei volumi (ovvero il daemon `vold`).
2. Cambiare le directory nel punto di avvio del DVD.
3. Cambiare le directory con la directory `modelrun`.
4. Passare alla directory della piattaforma di pertinenza.
5. Eseguire lo script di installazione `.bin`. Ad esempio:

```
./modelersolutionpublisherhpa64.bin -i console
```
6. Vengono visualizzati i dettagli di introduzione. Premere Invio per continuare.
7. Vengono visualizzate le informazioni sulle licenze. Leggere la licenza, digitare 1 per accettarne i termini e premere Invio per continuare.

8. Viene visualizzato un elenco delle lingue disponibili. Immettere il numero della versione di lingua da installare e premere Invio per continuare.
9. Viene richiesto di immettere il percorso di installazione. Per utilizzare la directory predefinita, `/usr/IBM/SPSS/ModelerSolutionPublisher<nn>` (dove `<nn>` è il numero di versione), premere Invio. Il Runtime verrà installato nella directory specificata.
10. Viene richiesto di confermare il percorso di installazione. Se il percorso è corretto, digitare `y` e premere Invio.
11. Viene visualizzato un riepilogo delle informazioni di preinstallazione per confermare le impostazioni specificate. Premere Invio per continuare.
12. Un messaggio indica che la routine di installazione è pronta per essere eseguita. Premere Invio per continuare.
13. Lo stato di esecuzione della routine è visibile sulla barra di avanzamento visualizzata. Al termine, premere Invio per uscire dal programma di installazione.

Configurazione di ODBC su UNIX

Per impostazione predefinita, il gestore dei driver DataDirect non è configurato perché IBM SPSS Modeler Solution Publisher Runtime utilizzi ODBC nei sistemi UNIX. Per configurare UNIX per il caricamento del gestore dei driver DataDirect, immettere i comandi seguenti (dove `sp_install_dir` è la directory d'installazione di Solution Publisher Runtime):

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

Risoluzione dei problemi di un'installazione

Firma digitale non valida rilevata all'installazione

I prodotti IBM SPSS Modeler utilizzano il certificato emesso da IBM per la firma digitale. In determinate circostanze potrebbe essere visualizzato il seguente errore quando si tenta di installare i prodotti SPSS Modeler:

Errore 1330. Impossibile installare un file necessario. La firma digitale del file CAB [nome file] non è valida...

Tutti gli utenti Windows

Viene visualizzato questo messaggio se si tenta di installare prodotti SPSS Modeler su una macchina sprovvista di collegamento a Internet e sulla quale non è presente il certificato corretto. Utilizzare la seguente procedura per correggere questo problema.

1. Fare clic su **OK** per accettare il messaggio.
2. Fare clic su **Annulla** per uscire dal programma di installazione.
3. Se la macchina su cui si sta tentando di installare il prodotto non è collegata a Internet, eseguire l'operazione seguente su una macchina con collegamento Internet e copiare il file `.cer` sulla macchina in cui si eseguirà l'installazione.
4. Da <http://www.verisign.com/support/roots.html>, scorrere verso il basso per individuare **VeriSign Class 3 Primary CA - G5** root Certification Authority (CA). Scaricare questo certificato CA radice e salvarlo come file `.cer`.
5. Fare doppio clic sul file `.cer`.
6. Nella scheda Generale, fare clic su **Installa certificato**.
7. Seguire le istruzioni all'interno di Importazione guidata certificati, mantenendo le opzioni predefinite e facendo clic su **Fine** al termine.
8. Riprovare l'installazione.

Avvio del Runtime

Dopo aver installato il Runtime, è possibile utilizzarlo per eseguire flussi che sono stati pubblicati da IBM SPSS Modeler utilizzando uno dei nodi di esportazione. Per ulteriori informazioni consultare l'argomento "Pubblicazione di flussi" a pagina 8. Avviare il Runtime dalla riga di comando, con opzioni indicanti il nome del file di immagine pubblicato e un file di parametri facoltativo da eseguire.

Per eseguire il Runtime, al prompt dei comandi digitare la riga seguente:

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

dove le voci di comando sono illustrate nella tabella seguente.

Tabella 5. Voci di comando runtime

| Voci di comando | Descrizione |
|------------------|--|
| <parameter-file> | è il nome di file del file di parametri pubblicato (facoltativo). |
| <options> | è un elenco separato da virgole di coppie option=value che specificano le opzioni di esecuzione per il runtime. Nota: Dopo la virgola non vi devono essere spazi vuoti. |
| <image file> | è il nome di file del file di immagine pubblicato (*.pim) da eseguire. |

Nota: quando si utilizzano lingue multibyte come il cinese o il giapponese in ambiente UNIX, è necessario specificare il codice di lingua da utilizzare. L'esempio seguente mostra i codici di lingua per il cinese semplificato:

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

in cui -p è la posizione dei file PIM e PAR.

Consultare Capitolo 3, "Esecuzione di flussi pubblicati", a pagina 11 per ulteriori informazioni.

Modifica della directory Temp

Alcune operazioni eseguite da IBM SPSS Modeler Solution Publisher Runtime potrebbero richiedere la creazione di file temporanei. Per impostazione predefinita, per la creazione di file temporanei in Runtime viene utilizzata la directory temporanea di sistema. È possibile modificare il percorso della directory temporanea, attenendosi alla procedura seguente.

Nota: è molto importante utilizzare la sintassi corretta. Prestare particolare attenzione agli spazi ed al percorso della directory.

Si supponga di utilizzare IBM SPSS Modeler Solution Publisher Runtime con i seguenti file pubblicati, *webtest.pim* e *webtest.par*, e di voler modificare la posizione dei file temporanei per utilizzare la directory *C:\published\temp*. Per modificare la posizione della directory temp utilizzata dall'eseguibile:

1. Dalla directory in cui sono salvati i file *.pim* e *.par*, eseguire il seguente comando:

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

Runtime utilizzerà la directory "*C:\published\temp*" come directory temporanea solo per quel processo specifico.

Disinstallazione di IBM SPSS Modeler Solution Publisher Runtime

Nella sezione seguente sono descritte le procedure per la disinstallazione di IBM SPSS Modeler Solution Publisher Runtime.

Procedura di disinstallazione su computer Windows

1. Nel Pannello di controllo di Windows, selezionare **Installazione applicazioni**.
2. Dall'elenco, selezionare **IBM SPSS Modeler Solution Publisher <nn>**.
3. Fare clic su **Rimuovi**.
4. Per rimuovere il Runtime, seguire le istruzioni visualizzate sullo schermo.

Nota: per rimuovere programmi, è necessario accedere al computer con i privilegi di amministratore.

Procedura di disinstallazione su computer UNIX

Rimuovere la directory di installazione, inclusi tutti i file installati.

Capitolo 5. API di IBM SPSS Embedded Predictive Modeling

L'API di IBM SPSS Embedded Predictive Modeling definisce una semplice interfaccia Java affinché si possa integrare le risorse utente progettate IBM SPSS Modeler con le proprie applicazioni in modalità incorporata. La documentazione completa in formato Javadoc è inclusa con IBM SPSS Modeler Solution Publisher per programmatori che utilizzano la libreria.

Prendere nota delle seguenti limitazioni quando si utilizza l'API EPM (Embedded Predictive Modeling):

- **Sono richieste etichette univoche di nodi origine e terminale.** L'API EPM utilizza le etichette dei nodi. Sebbene le etichette si trovano sempre in un contesto nodo Origine o Terminale, possono essere ambigue.
- **I nodi super non sono supportati come nodi origine o terminale.** Per fornire un'origine comune per i nodi Super in tutte le operazioni, essi non sono supportati come nodi Origine o Terminale
- **Limitazioni al nome del campo di input.** Alcuni caratteri non possono essere utilizzati in nomi campo.

Consultare le seguenti informazioni per una panoramica della funzionalità API. Per ulteriori dettagli, consultare il Javadoc incluso nel prodotto.

Calcolo del punteggio incorporato nel modello predittivo

Utilizzo: Utilizzato dall'applicazione per generare analisi predittive su richiesta man mano che entrano nuovi dati. L'applicazione deve gestire modelli aggiornati. Qualsiasi nodo di origine, nonché il nodo terminale del ramo di calcolo del punteggio, viene sostituito da input e output dell'applicazione in questo nodo di calcolo del punteggio.

Azione Prepara il ramo di calcolo del punteggio progettato nel proprio flusso IBM SPSS Modeler una volta per chiamate ripetute che generano rapidamente e in modo efficiente l'analisi predittiva dai dati di input gestiti dall'applicazione.

Aggiornamento del modello predittivo

Utilizzo: Eseguire la formazione iniziale degli algoritmi del modello nella progettazione del flusso utilizzando dati cronologici. Inoltre, "aggiornare" periodicamente gli algoritmi del modello predittivo configurato con i dati cronologici più recenti.

Azione: Esegue tutti i nodi builder del modello predittivo nel flusso IBM SPSS Modeler, che prepara l'algoritmo del modello predittivo come configurato utilizzando le tecniche di estrazione dati (Data mining). Aggiorna i nugget dell'applicatore di modelli collegati a ciascun nodo del builder di modelli. I parametri del flusso e le proprietà del nodo di origine/terminale possono essere utilizzati per implementare alcuni livelli delle dinamiche.

Valutazione del modello predittivo

Utilizzo: Utilizzato da un'applicazione per comunicare la precisione, la confidenza o altre valutazioni qualitative di un modello predittivo di apprendimento come progettato nel flusso IBM SPSS Modeler. Queste informazioni sono generalmente utilizzate per controllare la promozione di un modello predittivo appena aggiornato nel calcolo del punteggio dell'applicazione. L'obiettivo principale per cui l'aggiornamento modello restituisce un nuovo oggetto `PredictiveModel` è di consentire il confronto dei dati di valutazione tra il modello appena aggiornato e l'originale.

Azione: Esegue tutti i nodi builder del documento (principalmente i nodi di output non interattivi sulle tavolozze Output e Grafico IBM SPSS Modeler) nel flusso IBM SPSS Modeler che restituisce i documenti

risultanti. I parametri del flusso e le proprietà del nodo di origine/terminale possono essere utilizzati per implementare alcuni livelli delle dinamiche.

Esecuzione flusso del modello predittivo

Utilizzo: Tutti gli scenari di utilizzo che non si adattano ai modelli "refresh," "evaluation," o ad entrambi i modelli di calcolo del punteggio devono essere gestiti utilizzando l'esecuzione del flusso.

Azione: Esecuzione in base al controllo delle opzioni flusso IBM SPSS Modeler. Sia in modalità "tutti i nodi terminali", simile alla selezione mediante clic del pulsante a freccia verde in IBM SPSS Modeler Client, sia programmato nello script a livello flusso. I parametri del flusso e le proprietà del nodo di origine/terminale possono essere utilizzati per implementare alcuni livelli delle dinamiche.

Dati esterni di calcolo del punteggio nel modello predittivo

Utilizzo: Indicato comunemente come processo "batch" grazie alla modalità con cui un batch completo di dati di input viene elaborato.

Azione: Esegue il ramo di calcolo del punteggio come progettato. Gli input provenienti da nodi di origine di dati configurati e da analisi predittive generate risultano persistenti secondo il controllo del nodo terminale. I parametri del flusso e le proprietà del nodo di origine/terminale possono essere utilizzati per implementare alcuni livelli delle dinamiche.

Parametri flusso del modello predittivo e specifica delle proprietà del nodo di origine e del nodo terminale

Utilizzo: Un modo prestabilito consiste nel definire i parametri indicati nelle espressioni nei rami del flusso. Un altro punto di controllo consiste nel modificare determinare proprietà chiave di un nodo di origine o terminale per alterare il percorso file utilizzato, ecc.

Azione: determinati aspetti dell'esecuzione flusso IBM SPSS Modeler possono essere dinamicamente modificati da un'applicazione.

Nota: Le modifiche ai parametri devono essere effettuate prima di creare PredictiveScorer (consultare la descrizione del calcolo punteggio incorporato sopra riportata), mentre le proprietà dei nodi origine e terminale non hanno alcun significato per questo oggetto.

Capitolo 6. Riferimenti all'API della libreria di IBM SPSS Modeler Solution Publisher Runtime

Introduzione

L'API CLEMRTL consente di controllare l'esecuzione di flussi pubblicati dall'applicazione. In questa sezione sono descritte le funzioni API disponibili.

Di seguito sono riportati argomenti di interesse generale che riguardano l'utilizzo dell'API:

- L'API viene avviata al collegamento C per la massima compatibilità, ma la libreria ha dipendenze C++. In alcune piattaforme questo significa che può essere utilizzata solo con un linker che riconosce C++.
- Prima di utilizzare qualsiasi altra funzione della libreria, è necessario chiamare `clemrtl_initialize_ext()`.
- Il tipo `clemrtl_image_handle` è utilizzato come un identificatore di immagine.
- Ogni funzione restituisce un codice di indicatore di stato. La seguente tabella illustra i valori dell'indicatore di stato.

Tabella 6. I valori dell'indicatore di stato

| Risultato | Descrizione |
|---------------|---|
| CLEMRTL_OK | Esito positivo |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con informazioni aggiuntive sull'errore disponibili mediante la funzione <code>clemrtl_getErrorDetail()</code> |

Panoramica del processo API

Di seguito è riportata la struttura generale di un'applicazione che utilizza l'API:

1. Inizializzare la libreria con `clemrtl_initialize_ext()`.
2. Controllare e modificare le opzioni di esecuzione con `clemrtl_getOption()` e `clemrtl_setOption()`.
3. Aprire un'immagine con `clemrtl_openImage()` e ricevere un handle dell'immagine.
4. Controllare e modificare i parametri dell'immagine con `clemrtl_enumerateParameters()`, `clemrtl_getParameter()` e `clemrtl_setParameter()`.
5. Eseguire l'immagine con `clemrtl_execute()`. Se è necessario eseguire ripetutamente la stessa immagine senza modificarne i parametri, utilizzare prima `clemrtl_prepare()`.
6. Chiudere l'immagine con `clemrtl_closeImage()`.
7. Per annullare un'esecuzione in corso, utilizzare `clemrtl_interrupt()`.
8. Per recuperare informazioni sull'ultimo errore, utilizzare `clemrtl_getErrorDetail()`. Per ricevere messaggi di errore e altri messaggi diagnostici al loro arrivo, utilizzare `clemrtl_setReportHandler()`.

La libreria è stata progettata per l'utilizzo a thread multipli alle seguenti condizioni:

- Non è possibile chiamare contemporaneamente funzioni globali, ovvero quelle che non assumono un handle dell'immagine come argomento.
- Non è possibile chiamare contemporaneamente funzioni locali dell'immagine, ovvero quelle che assumono un handle dell'immagine come argomento, quando sono applicate allo stesso handle dell'immagine, con l'eccezione di `clemrtl_interrupt()`.

- È possibile chiamare contemporaneamente funzioni locali di immagine quando sono applicate a handle dell'immagine diversi.

In particolare, questo significa che thread separati possono preparare ed eseguire immagini contemporaneamente, a condizione che utilizzino handle dell'immagine diversi. Handle dell'immagine diversi possono fare riferimento alla stessa immagine (clemrtl_openImage() crea un nuovo handle ogni volta che viene chiamato, anche per la stessa immagine), tuttavia è necessario ricordarsi di modificare i parametri dell'immagine quando si eseguono istanze multiple della stessa immagine per reindirizzare l'output in base alle specifiche esigenze. Se due immagini eseguite contemporaneamente tentano di scrivere entrambe nello stesso file o tabella di database di output, i risultati saranno imprevedibili.

Funzioni API

Di seguito è riportato l'elenco completo delle funzioni esposte nell'API.

initialise

Questa funzione è obsoleta ed è equivalente a
`initialise_ext(flags, 0, 0);`

I nuovi programmi devono chiamare *initialise_ext*.

initialise_ext

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

Inizializza IBM SPSS Modeler Solution Publisher Runtime. *E' necessario chiamare questa funzione prima di qualsiasi altra funzione API.*

Tabella 7. Parametri *initialise_ext*

| Parametro | Descrizione |
|-----------|--|
| flags | Controlla alcuni aspetti del processo di inizializzazione. Il valore è costruito come un OR esclusivo bit per bit dei flag descritti di seguito. |
| arg_count | Numero di argomenti di inizializzazione aggiuntivi. Deve essere minore o uguale alla lunghezza dell'array args. |
| args | Argomenti di inizializzazione aggiuntivi. Gli argomenti sono descritti per tipo: <pre>typedef struct _clemrtl_init_arg { const char* name; const char* value; } clemrtl_init_arg;</pre> |

Il campo name è il nome di una proprietà di configurazione e il campo value è il valore corrispondente. Vengono riconosciute le proprietà di configurazione elencate nella tabella seguente.

Tabella 8. Proprietà di configurazione

| Nome | Valore |
|------------------------|---|
| installation_directory | Percorso completo alla cartella di installazione di IBM SPSS Modeler Solution Publisher. La cartella di installazione è quella contenente le cartelle bin e config. |

Restituisce uno dei codici di errore illustrati nella tabella seguente.

Tabella 9. Codici di errore

| Risultato | Descrizione |
|--------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Inizializzazione non riuscita, senza ulteriori dettagli disponibili. |

Il parametro `flag` fornisce il controllo di determinati aspetti del processo di inizializzazione. Il valore 0 (o `CLEMRTL_INIT_DEFAULTS`) specifica il comportamento predefinito, come indicato di seguito:

- Impostare il fuso orario predefinito chiamando `tzset()`.
- Impostare la locale `LC_NUMERIC` a "C".
- Impostare il nuovo gestore globale per generare un'eccezione quando la memoria è esaurita.

È possibile modificare questo comportamento specificando combinazioni degli indicatori elencati nella seguente tabella.

Tabella 10. Indicatore

| Indicatore | Descrizione |
|---|--|
| <code>CLEMRTL_INIT_NO_TZ</code> | Non impostare il fuso orario. |
| <code>CLEMRTL_INIT_NO_LOCALE</code> | Non modificare la locale. |
| <code>CLEMRTL_INIT_NO_NEW_HANDLER</code> | Non impostare il nuovo gestore. |
| <code>CLEMRTL_INIT_LOCAL_NEW_HANDLER</code> | Localizzare il nuovo gestore in ogni chiamata API. |

Per convertire correttamente tra numeri e stringhe, è necessario impostare la locale `LC_NUMERIC` a "C" (il formato di numero IBM SPSS Modeler utilizza le regole della locale "C"). È inclusa la conversione di numeri letti da e scritti in file di testo.

- Specificare `NO_LOCALE` se l'applicazione dipende dalla locale `LC_NUMERIC` e non richiede conversione tra numeri e stringhe.
- `NO_NEW_HANDLER` e `LOCAL_NEW_HANDLER` si escludono a vicenda. È necessario che `new` generi un'eccezione quando la memoria è esaurita.
- Specificare `NO_NEW_HANDLER` se l'applicazione imposta il proprio nuovo gestore oppure se si è certi che il compilatore/sistema di runtime generi un'eccezione per impostazione predefinita.

Specificare `LOCAL_NEW_HANDLER` se l'applicazione necessita che `new` restituisca un puntatore null quando la memoria è esaurita. *Nota:* questo utilizzo non è sicuro in ambienti a thread multipli.

getOption

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

Recupera il valore di un'opzione di esecuzione.

Tabella 11. parametri `getOption`

| Parametro | Descrizione |
|-------------------------|---|
| <code>name</code> | Nome dell'opzione. |
| <code>value</code> | Buffer che riceve il valore dell'opzione. |
| <code>value_size</code> | Dimensione del buffer del valore. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 12. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se il nome dell'opzione non è uno dei nomi di opzione di esecuzione elencati in precedenza, per esempio `memory_usage`, oppure se il buffer del valore non è sufficientemente grande per ricevere il valore, incluso un carattere di terminazione null. I valori delle opzioni sono sempre stringhe, anche quando l'interpretazione è numerica. Consultare Capitolo 3, "Esecuzione di flussi pubblicati", a pagina 11 per ulteriori informazioni.

setOption

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

Imposta il valore di un'opzione di esecuzione.

Tabella 13. Parametri setOption

| Parametro | Descrizione |
|-----------|----------------------|
| name | Nome dell'opzione. |
| value | Valore dell'opzione. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 14. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se il nome dell'opzione non è uno dei nomi di opzione di esecuzione elencati in precedenza. Per esempio, `memory_usage`. I valori delle opzioni sono sempre stringhe, anche quando l'interpretazione è numerica.

setLogFile

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

Reindirizza i messaggi di log da tutte le immagini aperte successivamente.

Tabella 15. Parametri setLogFile

| Parametro | Descrizione |
|-----------|--|
| mode | La modalità di registrazione deve essere uno dei valori riportati di seguito: CLEMRTL_NULL_LOG sopprime i messaggi. CLEMRTL_STDERROR_LOG invia i messaggi allo standard output degli errori. CLEMRTL_FILE_LOG invia i messaggi al file denominato dal parametro <code>log_file</code> . |

Tabella 15. Parametri setLogFile (Continua)

| Parametro | Descrizione |
|-----------|------------------------------------|
| log_file | Stringa che contiene il nome file. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 16. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se la modalità non è una delle modalità riconosciute oppure se è necessaria la registrazione in un file e il nome file non è stato specificato oppure non è valido.

openImage

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

Apri l'immagine di un flusso pubblicato.

Tabella 17. Parametri openImage

| Parametro | Descrizione |
|------------|--------------------------------|
| image_file | Il nome file dell'immagine. |
| param_file | Il nome file dei parametri. |
| handle | Riceve l'handle dell'immagine. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 18. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

Il file dei parametri può essere NULL se non è richiesto un file dei parametri separato. L'operazione non riesce se non si specifica il file di immagine, se il nome file non è valido oppure se il contenuto non è leggibile. Se l'operazione riesce, l'handle dell'immagine restituito può essere utilizzato per identificare l'istanza dell'immagine nelle successive chiamate API.

closeImage

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

Chiude un handle dell'immagine e ne libera le risorse.

Tabella 19. Parametri `closeImage`

| Parametro | Descrizione |
|---------------------|-------------------------|
| <code>handle</code> | L'handle dell'immagine. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 20. Codici di stato

| Risultato | Descrizione |
|----------------------------|--|
| <code>CLEMRTL_OK</code> | Operazione riuscita. |
| <code>CLEMRTL_FAIL</code> | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| <code>CLEMRTL_ERROR</code> | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se è in corso un'altra chiamata API sull'immagine, in particolare se l'immagine è ancora in esecuzione. Se l'operazione riesce, l'immagine viene chiusa e non è possibile utilizzare l'handle in nessun'altra chiamata API successiva, eccetto che in un'altra chiamata di `clemrtl_closeImage()`, che non produce effetti.

enumerateParameters

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

Applica una procedura di callback a ogni nome e valore di parametro di immagine.

Tabella 21. Parametri `enumerateParameters`

| Parametro | Descrizione |
|---------------------|--|
| <code>handle</code> | L'handle dell'immagine. |
| <code>proc</code> | La procedura di callback dei parametri. |
| <code>data</code> | Dati definiti dall'utente per il callback. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 22. Codici di stato

| Risultato | Descrizione |
|----------------------------|--|
| <code>CLEMRTL_OK</code> | Operazione riuscita. |
| <code>CLEMRTL_ERROR</code> | Operazione non riuscita, con ulteriori dettagli disponibili. |

Di seguito è riportato il tipo della procedura di callback:

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

Tabella 23. Parametri della procedura di callback

| Parametro | Descrizione |
|-------------------|--|
| <code>data</code> | Dati definiti dall'utente passati a <code>clemrtl_enumerateParameters()</code> . |

Tabella 23. Parametri della procedura di callback (Continua)

| Parametro | Descrizione |
|-----------|-----------------------|
| name | Nome del parametro. |
| value | Valore del parametro. |

La procedura viene applicata esattamente una sola volta a ogni parametro di immagine in ordine arbitrario.

getParameter

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

Recupera il valore di un parametro di immagine.

Tabella 24. parametri getParameter

| Parametro | Descrizione |
|------------|--|
| handle | L'handle dell'immagine. |
| name | Nome del parametro. |
| value | Buffer che riceve il valore del parametro. |
| value_size | Dimensione del buffer del valore. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 25. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se il nome di parametro non corrisponde al nome di qualsiasi parametro dell'immagine oppure se il buffer del valore non è sufficientemente grande da ricevere il valore, incluso un carattere di terminazione null. I nomi di parametro utilizzano il formato nome.attributo descritto in precedenza, per esempio, file0.nome e i valori di parametro sono sempre stringhe, anche quando l'interpretazione è numerica.

setParameter

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

Imposta il valore di un parametro di immagine.

Tabella 26. Parametri setParameter

| Parametro | Descrizione |
|-----------|-------------------------|
| handle | L'handle dell'immagine. |

Tabella 26. Parametri setParameter (Continua)

| Parametro | Descrizione |
|-----------|-----------------------|
| name | Nome del parametro. |
| value | Valore del parametro. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 27. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se l'handle dell'immagine non è valido oppure se il nome di parametro non corrisponde al nome di qualsiasi parametro dell'immagine. I nomi di parametro utilizzano il formato nome.attributo descritto in precedenza, per esempio, file0.nome e i valori di parametro sono sempre stringhe, anche quando l'interpretazione è numerica.

getFieldCount

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

Restituisce il numero di campi in un'origine di input o destinazione di output.

Tabella 28. Parametri getFieldCount

| Parametro | Descrizione |
|-------------|--|
| handle | L'handle dell'immagine. |
| key | Nome dell'input o dell'output da esaminare, come utilizzato nel file di parametri. La chiave può fare riferimento a un file o a un database. |
| field_count | Riceve il numero di campi. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 29. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

getFieldTypes

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

Restituisce i tipi di campo per un'origine di input o destinazione di output.

Tabella 30. parametri getFieldTypes

| Parametro | Descrizione |
|-------------|--|
| handle | L'handle dell'immagine. |
| chiave | Nome dell'input o dell'output da esaminare, come utilizzato nel file di parametri. La chiave può fare riferimento a un file o a un database. |
| field_count | Numero di campi da esaminare. Deve essere minore o uguale alla lunghezza dell'array field_types. |
| field_types | Array di lunghezza almeno pari a field_count che riceve i tipi di campo. Nell'array vengono copiati i tipi dei primi campi field_count dell'input o dell'output. Se field_count è maggiore del numero effettivo di campi, gli elementi aggiuntivi dell'array vengono lasciati non definiti. I valori dei tipi di campo vengono elencati nella tabella dei tipi di dati riportata di seguito. |

Tipi di dati

Il field_types deve essere uno dei tipi di dati riportati nella seguente tabella.

Tabella 31. Tipi di dati

| Tipo | Interpretazione | tDichiaratore 'C' tipico |
|-----------|---|--------------------------|
| STRING | Stringa di caratteri con terminazione NULL UTF-8. | const char* |
| INTEGER | Numero intero firmato a 32 bit. | int |
| LONG | Numero intero con segno a 64-bit. | long long |
| REAL | Numero a virgola mobile a 64 bit. | double |
| TIME | Numero intero firmato a 64 bit (secondi da mezzanotte). | long long |
| DATE | Numero intero firmato a 64 bit (secondi dalla mezzanotte del 01/01/1970). | long long |
| TIMESTAMP | Numero intero firmato a 64 bit (secondi dalla mezzanotte del 01/01/1970). | long long |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 32. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

setAlternativeInput

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

Sostituisce l'origine di input di un file con un'origine di input alternativa.

Tabella 33. Parametri `setAlternativeInput`

| Parametro | Descrizione |
|-------------|---|
| handle | L'handle dell'immagine. |
| key | Nome dell'origine di input originale, come utilizzata nel file di parametri. L'origine di input deve essere un file cosicché la chiave avrà sempre la forma "fileN" per alcuni interi $N \geq 0$. |
| field_count | Numero di campi nell'input. Il valore deve corrispondere esattamente al numero di campi nell'input del file originale. In caso contrario la chiamata non riuscirà. |
| field_types | Array di tipi di lunghezza pari almeno a field_count. Il valore field_types[i] è il tipo dell'i-esimo campo e deve essere uno dei valori elencati nella tabella dei tipi di dati riportata di seguito. I tipi devono essere compatibili con quelli dell'input del file originale. In caso contrario, la chiamata non riuscirà. |
| iterator | Funzione che produce dati di input alternativi. Si applica ai rispettivi argomenti come indicato di seguito: <pre>void** row = iterator(arg);</pre> <p>La funzione viene chiamata durante l'esecuzione (all'interno di una chiamata a <code>clemrtl_execute</code>) ed è chiamata una volta per ogni record di input. Un valore restituito NULL indica la fine dell'input. In questo caso, la funzione non viene più chiamata e l'esecuzione verrà terminata. In caso contrario, il risultato è un array di dati di lunghezza almeno pari a field_count, in cui row[i] fornisce il valore dell'i-esimo campo. Un valore può essere NULL. In caso contrario, deve essere un puntatore a un dato il cui tipo è determinato dal corrispondente field_types[i]. I puntatori devono rimanere validi fino alla chiamata successiva dell'iteratore o fino alla fine dell'esecuzione se l'esecuzione termina prematuramente.</p> |
| arg | Argomento opaco che viene passato all'iteratore a ogni chiamata. |

Tipi di dati

I field_types devono essere uno di questi tipi. Per ulteriori informazioni consultare l'argomento "getFieldTypes" a pagina 30.

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 34. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se i tipi di campo non sono compatibili con quelli dell'input del file originale oppure se il conteggio dei campi non corrisponde esattamente al numero di campi nell'input del file originale.

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);
```

Sostituisce la destinazione output di un file con una destinazione output alternativa.

Tabella 35. Parametri setAlternativeOutput

| Parametro | Descrizione |
|-------------|--|
| handle | L'handle dell'immagine. |
| key | Nome della destinazione di output originale, come utilizzata nel file di parametri. La destinazione di output deve essere un file cosicché la chiave avrà sempre la forma "fileN" per alcuni numeri interi $N \geq 0$. |
| field_count | Numero di campi nell'output. Il valore deve corrispondere esattamente al numero di campi nell'output del file originale. In caso contrario la chiamata non riuscirà. |
| field_types | Array di tipi di lunghezza pari almeno a field_count. Il valore field_types[i] è il tipo dell'i-esimo campo e deve essere uno dei valori elencati nella tabella dei tipi di dati riportata di seguito. I tipi devono essere compatibili con quelli dell'output del file originale. In caso contrario, la chiamata non riuscirà. |
| iterator | Funzione che consuma l'output dell'immagine. Si applica al rispettivo argomento e a una riga di dati come indicato di seguito: iterator(arg, row); La funzione viene chiamata durante l'esecuzione (all'interno di una chiamata a clemrtl_execute) ed è chiamata una volta per ogni riga di risultati prodotta dall'immagine. Un valore row di NULL indica la fine dell'output, dopo il quale la funzione non viene più chiamata. Un'applicazione non dovrebbe basarsi su questa chiamata finale e dovrebbe chiudere eventuali risorse esterne quando l'esecuzione viene terminata. In caso contrario, row è un array di dati di lunghezza almeno pari a field_count, in cui row[i] fornisce il valore dell'i-esimo risultato. Un valore può essere NULL. In caso contrario, deve essere un puntatore a un dato il cui tipo è determinato dal corrispondente field_types[i]. La funzione deve copiare gli eventuali valori di dati di cui ha bisogno perché la memoria potrebbe non venire conservata dopo che la chiamata viene restituita. |
| arg | Argomento opaco che viene passato all'iteratore a ogni chiamata. |

Tipi di dati

I field_types devono essere uno di questi tipi. Per ulteriori informazioni consultare l'argomento "getFieldTypes" a pagina 30.

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 36. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_FAIL | Operazione non riuscita, senza ulteriori dettagli disponibili. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se i tipi di campo non sono compatibili con quelli dell'output del file originale oppure il se conteggio dei campi non corrisponde esattamente al numero di campi nell'output del file originale.

execute

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

Esegue un'immagine.

Tabella 37. Parametri di esecuzione

| Parametro | Descrizione |
|-----------|-------------------------|
| handle | L'handle dell'immagine. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 38. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

Se l'immagine non è stata preparata, viene preparata inizialmente utilizzando i valori di parametro correnti. L'operazione non riesce se l'handle dell'immagine non è valido oppure se si verifica un errore durante la preparazione o l'esecuzione. La chiamata non restituisce finché l'esecuzione non è completa.

prepare

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

Prepara un'immagine per l'esecuzione.

Tabella 39. preparare i parametri

| Parametro | Descrizione |
|-----------|-------------------------|
| handle | L'handle dell'immagine. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 40. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non riesce se l'handle dell'immagine non è valido oppure se si verifica un errore durante la preparazione.

È necessario preparare un'immagine prima della sua esecuzione. Durante la preparazione di un'immagine vengono bloccati i valori di parametro nell'immagine. Se non è stata già preparata, `clemrtl_execute()` prepara un'immagine automaticamente per ogni chiamata; utilizzando `clemrtl_prepare()` per preparare un'immagine è utile se l'immagine deve essere eseguita più volte con gli stessi valori di parametro e può migliorare sensibilmente le prestazioni. Dopo che un'immagine è stata preparata, le successive modifiche apportate ai valori di parametro vengono ignorate. Utilizzare nuovamente `clemrtl_prepare()` per aggiornare l'immagine con i nuovi valori di parametro.

interrupt

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

Termina l'esecuzione in corso di un'immagine.

Tabella 41. Parametri di `interrupt`

| Parametro | Descrizione |
|-----------|-------------------------|
| handle | L'handle dell'immagine. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 42. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

L'operazione non ha effetto se l'handle dell'immagine non è valido oppure se l'immagine non è in esecuzione.

Questa funzione è sicura per chiamare contemporaneamente con un'altra chiamata API sullo stesso handle dell'immagine.

getErrorDetail

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

Restituisce informazioni dettagliate sull'ultimo errore che si è verificato in un'immagine.

Tabella 43. Parametri `getErrorDetail`

| Parametro | Descrizione |
|-----------|--|
| handle | L'handle dell'immagine. |
| severity | Riceve il codice di gravità come carattere singolo: I: informazione W: avviso (warning) E: errore X: errore di sistema |
| code | Riceve il numero di errore. |
| text | Buffer che riceve il testo del messaggio. |
| text_size | Dimensione del buffer del testo. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 44. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

I risultati saranno inaffidabili se l'handle dell'immagine non è valido. Per adattarlo al buffer del testo, il testo del messaggio viene troncato all'occorrenza.

Se una chiamata API non riesce con CLEMRTL_ERROR e l'errore non è correlato a un handle dell'immagine specifico, passando 0 come handle dell'immagine verranno recuperati i dettagli dell'ultimo errore non specifico di un'immagine, ma il risultato non è affidabile in ambienti a thread multipli.

setReportHandler

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

Installa una procedura di callback di report per un'immagine.

Tabella 45. parametri setReportHandler

| Parametro | Descrizione |
|-----------|--|
| handle | L'handle dell'immagine. |
| proc | La procedura di callback del report. |
| data | Dati definiti dall'utente per il callback. |

Restituisce uno dei codici di stato illustrati nella tabella seguente.

Tabella 46. Codici di stato

| Risultato | Descrizione |
|---------------|--|
| CLEMRTL_OK | Operazione riuscita. |
| CLEMRTL_ERROR | Operazione non riuscita, con ulteriori dettagli disponibili. |

Di seguito è riportato il tipo della procedura di callback:

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

Tabella 47. Parametri della procedura di callback

| Parametro | Descrizione |
|-----------|---|
| data | Dati definiti dall'utente passati a clemrtl_setReportHandler(). |

Tabella 47. Parametri della procedura di callback (Continua)

| Parametro | Descrizione |
|-----------|---|
| severity | Il codice di gravità come carattere singolo: I: informazione W: avviso (warning) E: errore X: errore di sistema |
| code | Il numero del messaggio. |
| text | Il testo del messaggio. |

La procedura è applicata a ogni messaggio al suo arrivo. Impostando la procedura su NULL, qualsiasi gestore esistente installato nell'immagine viene rimosso.

Note

Queste informazioni sono state preparate per prodotti e servizi offerti in tutto il mondo.

IBM può non offrire i prodotti, i servizi o le funzioni presentati in questo documento in altri paesi. Consultare il rappresentante locale IBM per le informazioni sui prodotti e servizi attualmente disponibili nella propria zona. Qualsiasi riferimento ad un prodotto, programma o servizio IBM non implica o intende dichiarare che solo quel prodotto, programma o servizio IBM può essere utilizzato. In sostituzione a quelli forniti da IBM, è possibile utilizzare prodotti, programmi o servizi funzionalmente equivalenti che non comportino violazione dei diritti di proprietà intellettuale o di altri diritti IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM può avere applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La consegna del presente documento non conferisce alcuna licenza rispetto a questi brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

IBM Director of Licensing
IBM Europe
Schoenaicher Str.220
D-7030 Boeblingen
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS) contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

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

Il seguente paragrafo non è valido per il Regno Unito o per tutte le nazioni le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute: L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA", SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITÀ ED IDONEITÀ AD UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia ad alcune garanzie espresse o implicite in determinate transazioni; pertanto, la presente dichiarazione potrebbe non essere sempre applicabile.

Le presenti informazioni possono includere imprecisioni tecniche o errori tipografici. Le modifiche periodiche apportate alle informazioni contenute in questa pubblicazione verranno inserite nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto nel manuale in qualsiasi momento e senza preavviso.

Qualsiasi riferimento nelle presenti informazioni a siti Web non IBM viene fornito esclusivamente per facilitare la consultazione e non rappresenta in alcun modo un'approvazione o sostegno da parte nostra di tali siti Web. I materiali disponibili sui siti Web non fanno parte di questo prodotto IBM e l'utilizzo di questi è a discrezione dell'utente.

IBM può utilizzare o distribuire le informazioni fornite in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

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

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, l'addebito di un canone.

Il programma concesso in licenza descritto nel presente documento e tutto il materiale concesso in licenza disponibile sono forniti da IBM in base ai termini dell'IBM Customer Agreement, dell'IBM International Program License Agreement o di qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questa pubblicazione sono stati determinati in un ambiente controllato. Di conseguenza, i risultati ottenuti con sistemi operativi diversi possono variare in modo significativo. Alcune misurazioni potrebbero essere state effettuate su sistemi in corso di sviluppo e non c'è garanzia che tali misurazioni coincidano con quelle effettuate sui sistemi comunemente disponibili. Inoltre, alcune misurazioni potrebbero essere stime elaborate tramite l'estrapolazione. I risultati effettivi potrebbero variare. Gli utenti di questo documento devono verificare i dati applicabili al proprio ambiente specifico.

Le informazioni relative a prodotti non IBM sono ottenute dai fornitori di quei prodotti, dagli annunci pubblicati e da altre fonti disponibili al pubblico. IBM non ha testato tali prodotti e non può confermarne l'accuratezza delle prestazioni, la compatibilità o qualsiasi altra dichiarazione relativa a prodotti non IBM. Eventuali domande in merito alle funzionalità dei prodotti non IBM vanno indirizzate ai fornitori di tali prodotti.

Tutte le dichiarazioni relative all'orientamento o alle intenzioni future di IBM sono soggette a modifica o a ritiro senza preavviso e rappresentano unicamente mete ed obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali. Per fornire una descrizione il più possibile esaustiva, gli esempi includono nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi utilizzati da gruppi aziendali realmente esistenti è puramente casuale.

Se si visualizza una copia elettronica di queste informazioni, è possibile che le illustrazioni a colori e le fotografie non vengano visualizzate.

Marchi

IBM, il logo IBM e ibm.com sono marchi o marchi registrati di International Business Machines Corp., registrati in numerose giurisdizioni del mondo. I nomi di altri prodotti e servizi potrebbero essere marchi di IBM o di altre società. Un elenco aggiornato di marchi IBM è disponibile sul Web nella sezione "Copyright and trademark information" all'indirizzo www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium sono marchi o marchi registrati di Intel Corporation o relative controllate negli Stati Uniti e altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Microsoft, Windows, Windows NT e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

UNIX è un marchio registrato di Open Group negli Stati Uniti e/o in altri paesi.

Java e tutti i marchi e i logo relativi a Java sono marchi commerciali o marchi registrati di Oracle e/o delle sue affiliate.

Altri nomi di prodotti e servizi possono essere marchi commerciali di IBM o di altre aziende.

Indice analitico

A

- apertura
 - flussi 27
- API
 - cenni generali 13, 23
 - codici di errore 23
 - codici di stato 23
 - conteggio campi 30
 - input alternativo 31
 - opzioni di esecuzione 25, 26
 - output alternativo 33
 - riferimenti 23
 - setReportHandler 36
 - tipi di campo 30

C

- chiusura
 - flussi 27
- closeImage
 - panoramica del processo API 23
 - riferimenti alle funzioni API 27
- codici di errore 23
- codici di stato 23

D

- directory temporanea
 - IBM SPSS Modeler Solution Publisher Runtime 18
- disinstallazione
 - IBM SPSS Modeler Solution Publisher Runtime 18
- distribuzione 7, 21
- documentazione 3

E

- enumerateParameters
 - panoramica del processo API 23
 - riferimenti alle funzioni API 28
- EPM
 - API 21
- esecuzione di flussi
 - utilizzo di IBM SPSS Modeler Solution Publisher 11, 12
- esempi
 - cenni generali 5
 - Guida alle applicazioni 3
- esempi di applicazioni 3
- execute
 - panoramica del processo API 23
 - riferimenti alle funzioni API 34

F

- file .par 8
- file .pim 8
- file di log 26

- funzioni API
 - closeImage 23, 27
 - enumerateParameters 23, 28
 - execute 23, 34
 - getErrorDetail 23, 35
 - getFieldCount 23, 30
 - getFieldTypes 23, 30
 - getOption 23, 25
 - getParameter 23, 29
 - inizializzazione 23, 24
 - interrupt 23, 35
 - openImage 23, 27
 - prepare 23, 34
 - setAlternativeInput 23, 31
 - setAlternativeOutput 23, 33
 - setLogFile 23, 26
 - setOption 23, 26
 - setParameter 23, 29
 - setReportHandler 23
- fuso orario
 - impostazione tramite l'API 24

G

- getErrorDetail
 - panoramica del processo API 23
 - riferimenti alle funzioni API 35
- getFieldCount
 - panoramica del processo API 23
 - riferimenti alle funzioni API 30
- getFieldTypes
 - panoramica del processo API 23
 - riferimenti alle funzioni API 30
- getOption
 - panoramica del processo API 23
 - riferimenti alle funzioni API 25
- getParameter
 - panoramica del processo API 23
 - riferimenti alle funzioni API 29

I

- IBM SPSS Embeddable Predictive Analytics
 - API 21
- IBM SPSS Modeler 1
 - documentazione 3
- IBM SPSS Modeler Server 1
- IBM SPSS Modeler Solution Publisher 7, 8, 21
 - cenni generali 7
 - integrazione con altre applicazioni 13
- IBM SPSS Modeler Solution Publisher Runtime
 - avvio 18
 - directory temp 18
 - disinstallazione 18
 - Installazione 15
 - requisiti di sistema 15, 16

- initialise
 - indicatori 24
 - riferimenti alle funzioni API 24
- initialise_ext
 - indicatori 24
 - riferimenti alle funzioni API 24
- initialize
 - panoramica del processo API 23
- interrupt
 - panoramica del processo API 23
 - riferimenti alle funzioni API 35

L

- locale
 - impostazione tramite l'API 24

N

- nodi output 7, 21
- nodo IBM SPSS Modeler Solution Publisher 21
- nodo Pubblica 21

O

- openImage
 - panoramica del processo API 23
 - riferimenti alle funzioni API 27

P

- parametri
 - file di parametri API 27
 - parametri di immagini API 28, 29
 - parametri per esecuzione del flusso 12
 - procedure di callback API 28
- prepare
 - panoramica del processo API 23
 - riferimenti alle funzioni API 34
- pubblicazione di flussi
 - IBM SPSS Modeler Solution Publisher 7, 8

R

- Runtime
 - avvio 18
 - directory temp 18
 - disinstallazione 18
 - installazione su computer UNIX 16
 - installazione su computer Windows 15
 - opzioni 11
- Runtime programming library (CLEMRTL) 13

S

- segnalazione di errori
 - referimenti alle funzioni API 26, 35, 36
- setAlternativeInput
 - panoramica del processo API 23
 - referimenti alle funzioni API 31
- setAlternativeOutput
 - panoramica del processo API 23
 - referimenti alle funzioni API 33
- setLogFile
 - panoramica del processo API 23
 - referimenti alle funzioni API 26
- setOption
 - panoramica del processo API 23
 - referimenti alle funzioni API 26
- setParameter
 - panoramica del processo API 23
 - referimenti alle funzioni API 29
- setReportHandler
 - panoramica del processo API 23
 - referimenti alle funzioni API 36

U

- UNIX
 - installazione di IBM SPSS Modeler Solution Publisher Runtime 16

W

- Windows
 - installazione di IBM SPSS Modeler Solution Publisher Runtime 15



Stampato in Italia