

IBM SPSS Collaboration and Deployment
Services - Essentials for Python
Versione 8 Release 2

Guida per lo sviluppatore



Nota

Prima di utilizzare queste informazioni e il prodotto supportato, leggere attentamente le informazioni riportate in [“Informazioni particolari” a pagina 67](#).

Informazioni sul prodotto

Questa edizione si applica alla versione 8, release 2, modifica 2 di IBM® SPSS Collaboration and Deployment Services e a tutte le successive release e modifiche fino a quando diversamente indicato nelle nuove edizioni.

© Copyright International Business Machines Corporation 2000, 2020.

Indice

Capitolo 1. IBM SPSS Collaboration and Deployment Services - Essentials for Python	1
Panoramica.....	1
Installazione.....	1
Funzioni obsolete.....	1
Capitolo 2. Creazione di script dalla riga comandi.....	3
Parole chiave globali.....	3
Connessioni al repository.....	4
Script del repository dei contenuti.....	5
Parole chiave.....	5
Operazioni Repository di contenuti.....	6
Funzioni di gestione dei processi.....	22
Parole chiave.....	23
Operazioni di gestione dei processi.....	23
Capitolo 3. Il modulo PESImpl.....	27
API repository di contenuti.....	28
Metodi.....	28
Classi Wrapper.....	56
API di gestione dei processi.....	59
Metodi.....	59
Classi wrapper.....	62
Script di esempio.....	66
Informazioni particolari.....	67
Considerazioni relative alla politica sulla privacy	68
Marchi.....	69

Capitolo 1. IBM SPSS Collaboration and Deployment Services - Essentials for Python

Panoramica

IBM SPSS Collaboration and Deployment Services fornisce una struttura di script con una serie di API che gli utenti avanzati e gli amministratori possono utilizzare per scrivere routine indipendenti o lavori batch che combinano una serie di routine per eseguire operazioni con lavori e oggetti del repository. Ciò può semplificare notevolmente le attività di massa, incluso quanto segue:

- Modifica delle autorizzazioni di sicurezza di un gruppo di utenti di grandi dimensioni
- Assegnazione di etichette o la rimozione di un'etichetta da un gran numero di cartelle o file
- Il caricamento o lo scaricamento di un gran numero di cartelle o i file

Il framework include la capacità di eseguire le attività dalla riga di comando, oltre a un'API per interagire con il IBM SPSS Collaboration and Deployment Services Repository all'interno del proprio codice Python.

Per informazioni generali su Python, un linguaggio di programmazione dinamico orientato agli oggetti, vedere il [sito Web di Python](#).

Installazione

Il framework di creazione script può essere installato su piattaforme Windows e UNIX. La piattaforma di creazione script è indipendente dalla piattaforma utilizzata dal repository a cui accede la funzione di creazione script.

Ad esempio, un repository in esecuzione sulla piattaforma Windows può essere richiamato da funzioni di script in esecuzione sulla piattaforma UNIX.

Per le istruzioni di installazione, consultare la Guida all'installazione di IBM SPSS Collaboration and Deployment Services - Essentials for Python.

Funzioni obsolete

Se si sta eseguendo la migrazione da una release precedente di IBM SPSS Collaboration and Deployment Services, è necessario essere a conoscenza delle diverse funzioni che sono state rese obsolete dall'ultima versione.

Se una funzione è obsoleta, IBM Corp. potrebbe rimuovere questa funzionalità in una release successiva del prodotto. Gli investimenti futuri saranno concentrati sulla funzione strategica elencata sotto Azione di migrazione consigliata. Generalmente, una funzione non è dichiarata obsoleta fino a quando non viene fornita un'alternativa equivalente.

Nessuna funzione è obsoleta in questa release. A scopo di riferimento, la seguente tabella indica le funzioni obsolete nelle recenti versioni precedenti del prodotto. Dove possibile, la tabella indica anche l'azione di migrazione consigliata.

<i>Tabella 1. Funzioni obsolete nelle versioni precedenti</i>	
Funzione obsoleta	Azione di migrazione consigliata
Provider di sicurezza: Active Directory con sostituzione locale, che supporta gruppi estesi ed utenti consentiti	Utilizzare il provider di sicurezza Active Directory con tutti i gruppi necessari aggiunti

Tabella 1. Funzioni obsolete nelle versioni precedenti (Continua)

Funzione obsoleta	Azione di migrazione consigliata
IBM SPSS Collaboration and Deployment Services Enterprise View	Utilizzare la funzione Vista dati analitici
IBM SPSS Collaboration and Deployment Services Enterprise View Driver	Utilizzare la funzione Vista dati analitici
File Scenario	I file Scenario (.scn) non sono più supportati. I nodi di origine Enterprise View non possono essere modificati in Deployment Manager. I file scenario precedenti possono essere modificati nel client IBM SPSS Modeler e salvati nuovamente come file di flusso. Sarà inoltre necessario eliminare le configurazioni di calcolo del punteggio che utilizzavano un file scenario e salvarle nuovamente sulla base di un file di flusso.
Installazione Web per IBM SPSS Deployment Manager	Utilizzare il programma di installazione autonomo
BIRT Report Designer for IBM SPSS	Nessuna
Visualizzatore BIRT Report Designer for IBM SPSS	Nessuna
IBM SPSS Collaboration and Deployment Services Portlet	Utilizzare direttamente IBM SPSS Collaboration and Deployment Services Deployment Portal oppure utilizzare le API dei servizi Web
IBM SPSS Collaboration and Deployment Services Web Part	Utilizzare direttamente IBM SPSS Collaboration and Deployment Services Deployment Portal oppure utilizzare le API dei servizi Web
API Servizio di calcolo del punteggio V1	API Servizio di calcolo del punteggio V2
Servizio pianificazione server	Nessuna
Servizio gestione report	Nessuna
Operazione login di Servizio di autenticazione	Operazione doLogin Servizio di autenticazione
Operazione search di Servizio di ricerca	Operazione search2.5 di Servizio di ricerca
Jar del client dei servizi Web SPSS AXIS/Castor	Utilizzare gli strumenti forniti con JRE (Java Runtime Environment), IDE (Integrated Development Environment) o Eclipse WTP (Web Tools Platform)
Funzione API clemrtl_setLogFile()	Nessuna

Capitolo 2. Creazione di script dalla riga comandi

È possibile utilizzare il file Python `CADSTool.py` dalla riga comandi per modificare le risorse memorizzate in IBM SPSS Collaboration and Deployment Services Repository.

La sintassi generale utilizzata per richiamare le operazioni di creazione script di IBM SPSS Collaboration and Deployment Services dalla riga comandi è:

```
python CADSTool.py <Operation>  
<Keywords>
```

Dove:

- `<Operation>` indica la funzione da richiamare
- `<Keywords>` definisce le coppie parola chiave/valore utilizzate come parametri di input per la funzione

Parole chiave globali

La Tabella 2 a pagina 3 elenca le parole chiave supportate da tutte le funzioni di creazione script di IBM SPSS Collaboration and Deployment Services. La seconda colonna elenca le versioni brevi facoltative delle parole chiave. Le parole chiave sono sensibili al maiuscolo/minuscolo.

Parola chiave	Versione breve facoltativa	Uso
<code>--user</code>	<code>-u</code>	<p>Il nome utente per la connessione al server del repository. Il valore deve includere un prefisso che indica il provider di sicurezza se l'utente non proviene dal provider nativo. Sono validi i valori di prefisso riportati di seguito:</p> <ul style="list-style-type: none">• <code>Native</code> per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito.• <code>AD_<name></code> per Active Directory, dove <code><name></code> corrisponde al nome del provider di sicurezza all'interno del sistema• <code>ADL_<name></code> per Active Directory con sovrascrittura locale, dove <code><name></code> corrisponde al nome del provider di sicurezza all'interno del sistema• <code>ldap_<name></code> per OpenLDAP, dove <code><name></code> corrisponde al nome del provider di sicurezza all'interno del sistema <p>Inserire una barra ed il nome utente dopo il prefisso. Per i provider Active Directory, includere il dominio nel prefisso. Ad esempio, per l'utente <code>icrod</code> nel dominio <code>MYDOMAIN</code> dell'istanza Active Directory <code>AD_SPSSAD</code>, il valore è <code>AD_SPSSAD/MYDOMAIN/icrod</code>. Se l'utente <code>icrod</code> si trova nel provider OpenLDAP <code>SPSSLDAP</code>, il valore è <code>ldap_SPSSLDAP/icrod</code>.</p>
<code>--password</code>	<code>-p</code>	La password per la connessione al server del repository

Tabella 2. Parole chiave globali (Continua)

Parola chiave	Versione breve facoltativa	Uso
--host	-q	Il nome dell'host/server in cui è installato il repository
--port	-o	Il numero di porta del server del repository
--ssl		Indica che il server del repository utilizza il protocollo SSL (secure sockets layer) per codificare le comunicazioni. Se si utilizza questa parola chiave, il server del repository deve essere configurato per SSL. Per ulteriori informazioni, consultare la documentazione dell'amministratore.
--server_url	-S	URL completo del server del repository. Utilizzare questa parola chiave se l'URL del server comprende una root di contesto personalizzata. I valori per le parole chiave host, port e ssl non sono necessarie se viene specificato l'URL del server.
--useDefault	-z	Utilizza le informazioni sulla connessione al server definite nel file Authorization.properties
--help	-h	Informazioni della guida del modulo di creazione script

Connessioni al repository

È necessario specificare l'identificativo utente IBM SPSS Collaboration and Deployment Services Repository, la password e le informazioni sul server del repository alla fine di ciascun comando.

Per fornire tali informazioni sulla connessione, è possibile utilizzare i seguenti metodi:

- Utilizzo delle parole chiave, come negli esempi riportati di seguito:

```
--user user --password
password --host host --port
port
```

```
--user
user --password password --server_url
url
```

- Mediante il file Authorization.properties, dove il comando contiene un parametro --useDefault (oppure la versione breve -z). Questo approccio richiama le informazioni sulla connessione dal file Authorization.properties, che si trova in *Scripting folder\Lib\site-packages\config\Authorization.properties*. Utilizzare un editor di testo per modificare i seguenti valori nel file in modo che corrispondano alle impostazioni del repository:

```
#
Authorization Information
user=admin
password=pwd
host=yourhost
port=80
```

In alternativa, è possibile utilizzare la parola chiave server_url nel file delle proprietà.

```
#
Authorization Information
user=admin
password=pwd
server_url=http://yourhost:80/context_root
```

I parametri passati mediante la riga comandi hanno sempre precedenza. Ad esempio, se vengono forniti i parametri `--user` e `--password` sulla riga comandi e viene fornito anche il parametro `--useDefault` o `-z`, vengono utilizzati l'utente e la password dalla riga comandi e l'host e la porta vengono richiamati dal file `Authorization.properties`. In alternativa, se l'utente, la password, l'host e la porta vengono tutti forniti nella riga comandi ma viene utilizzato anche il parametro `--useDefault` o `-z`, `--useDefault` viene ignorato e vengono utilizzate solo le informazioni della riga comandi.

Per tutte le API descritte, la sintassi e gli esempi utilizzano il parametro `-z` per utilizzare il numero minimo di parametri richiesti.

Script del repository dei contenuti

Lo script del repository dei contenuti consente di utilizzare le risorse del repository, quali ad esempio file e cartelle. Quest'area comprende le seguenti funzionalità:

- Creazione ed eliminazione di cartelle
- Caricamento e scaricamento dei file
- Esportazione ed importazione delle cartelle
- Gestione delle etichette, della sicurezza e dei metadati

Questa sezione descrive l'utilizzo da parte della riga comandi Python degli script per le funzioni del repository. Ciascuna operazione contiene informazioni dettagliate sulla sintassi, un esempio ed i messaggi previsti.

Parole chiave

Tabella 3 a pagina 5 elenca le parole chiave supportate per le funzioni di repository. La seconda colonna elenca le versioni facoltative, abbreviate delle parole chiave.

Importante: le parole chiave sono sensibili al maiuscolo/minuscolo.

Parola chiave	Versione breve facoltativa	Utilizzo
<code>--source</code>	<code>-s</code>	Il file origine o il percorso cartella
<code>--target</code>	<code>-t</code>	Il percorso della cartella di destinazione
<code>--version</code>	<code>-v</code>	La versione di un file
<code>--principal</code>	<code>-r</code>	L'utente a cui deve essere concessa l'autorizzazione
<code>--permission</code>	<code>-n</code>	Il tipo di autorizzazione (come ad esempio lettura, scrittura, modifica, autorizzazione ad eliminare)
<code>--label</code>	<code>-l</code>	L'etichetta da assegnare ad una versione di un file
<code>--criteria</code>	<code>-c</code>	Il criterio di ricerca per ricercare gli attributi di metadati dei file o cartelle.
<code>--author</code>	<code>-a</code>	Il nome autore per un file o cartella-
<code>--description</code>	<code>-d</code>	La descrizione di un file o cartella
<code>--title</code>	<code>-i</code>	Il titolo per un file o cartella
<code>--expirationDate</code>	<code>-q</code>	La data di scadenza di un file o cartella
<code>--expirationStartDate</code>		La data di inizio scadenza di un file o cartella
<code>--expirationEndDate</code>		La data finale di scadenza di un file o cartella

Tabella 3. Parole chiave per le API del repository (Continua)

Parola chiave	Versione breve facoltativa	Utilizzo
--keyword	-k	La parola chiave per un file o cartella
--cascade	-x	Indica che le impostazioni di sicurezza per una cartella devono propagarsi alle sottocartelle e file.
--provider	-f	Il provider di sicurezza utilizzato per recuperare i principal
--createVersion	-b	Indica che deve essere creata una nuova versione di un file.
--contentLanguage	-g	La lingua contenuto per un file o cartella.
--topic		L'argomento assegnato ad un file o cartella. È possibile immettere più valori come un --topic "topic1;topic2"
--modifiedBy		L'utente che ha modificato un file o cartella.
--mimeType		Il tipo mime di un file.
--createdBy		L'utente che ha creato un file o cartella.
--submittedHierarchy		Indica se ricercare la cartella <i>Lavori inviati</i>
--propertyName		Il nome di una proprietà personalizzata.
--customProperty		La coppia valore-nome di una proprietà personalizzata da aggiornare.
--propertyName		Il nome della proprietà personalizzata per recuperare i valori valido per

Per tutte le operazioni che accettano le informazioni relative all'etichetta e alla versione, l'utente dovrebbe specificare un'etichetta o una versione, ma non entrambe. Se, per un determinato file, non viene specificata nessuna versione o etichetta, viene utilizzata la versione più recente.

Operazioni Repository di contenuti

Operazione advanceSearch

Cerca file e cartelle nel repository, in base a diversi parametri.

Sintassi

```
python CADSTool.py advanceSearch --author <author>
--title <title> --description <description>
--createdBy <createdBy> --modifiedBy <modifiedBy>
--keyword <keyword> --label <label>
--topic <topic>
--uri <uri> --parentURI <parentURI>
--expirationStartDate <expirationStartDate>
--expirationEndDate <expirationEndDate>
--createdStartDate <createdStartDate>
--createdEndDate <createdEndDate>
--objectModifiedStartDate <objectModifiedStartDate>
--objectModifiedEndDate <objectModifiedStartDate>
--versionModifiedStartDate <versionModifiedStartDate>
--versionModifiedEndDate <versionModifiedEndDate>
--submittedHierarchy -z
```

Dove:

- <author> è il nome dell'autore.
- <title> è il titolo del file/cartella.


```
python CADSTool.py advanceSearch --uri "spsscr:///?id=a010a37ba5992bb00000127b0f952f945be" -z
```

Messaggi

Quando si utilizza questa API potrebbero essere visualizzati i seguenti messaggi:

- Quando l'API viene correttamente completata, viene visualizzato un elenco di file e cartelle che corrispondono ai criteri di ricerca. Vengono solitamente inseriti i nomi file con il relativo percorso completo e le versioni.
- Error searching files and folders
- error: no such option:<option>

Operazione applySecurity

Imposta l'ACL (access control list) di sicurezza per un file o una cartella nel repository.

Sintassi

```
python CADSTool.py applySecurity --source "<source>" --principal "<principal>"  
--permission "<permission>" --provider "<provider>" --cascade -z
```

Dove:

- *<source>* è il percorso completo IBM SPSS Collaboration and Deployment Services Repository del file o cartella in cui applicare l'ACL di sicurezza. Questo parametro è obbligatorio.
- *<principal>* è l'utente (ad esempio *admin*) da applicare al file o alla cartella specificati come parte dell'ACL. Questo parametro è obbligatorio.
- *<permission>* è il tipo di autorizzazione da applicare al file o alla cartella specificati (come la lettura, la scrittura, la modifica, la cancellazione o il proprietario). Questo parametro è obbligatorio.
- *<provider>* è il provider di sicurezza da utilizzare per il richiamo delle informazioni sugli utenti (principal). Questo parametro è facoltativo. I valori validi da includere:
 - Native per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito.
 - AD_*<name>* per Active Directory, dove *<name>* corrisponde al nome del provider di sicurezza all'interno del sistema
 - ADL_*<name>* per Active Directory con sovrascrittura locale, dove *<name>* corrisponde al nome del provider di sicurezza all'interno del sistema
 - ldap_*<name>* per OpenLDAP, dove *<name>* corrisponde al nome del provider di sicurezza all'interno del sistema
- `--cascade` viene utilizzato quando s'impone la sicurezza in una cartella, per propagare le impostazioni di sicurezza a tutti i file e le sottocartelle all'interno della cartella specificata. Questo parametro è facoltativo.

Esempi

Il seguente esempio applica la sicurezza ad una cartella:

```
python CADSTool.py applySecurity --source "/Projects/" --principal "icrod"  
--permission "READ" --provider "Native" -z
```

Il seguente esempio applica la sicurezza ad una cartella ed a tutti i file e le cartelle in essa contenuti:

```
python CADSTool.py applySecurity --source "/Projects/" --principal "icrod"  
--permission "READ" --provider "Native" --cascade -z
```

Il seguente esempio applica la sicurezza ad una cartella di un'entità principale in un provider di sicurezza Active Directory denominato SPSSAD:

```
python CADSTool.py applySecurity --source "/Projects" --principal "ICrod (MYDOMAIN)"
--permission "Write" --provider "AD_SPSSAD" -z
```

Il seguente esempio applica la sicurezza ad una cartella di un'entità principale in un provider di sicurezza OpenLDAP denominato LDAP:

```
python CADSTool.py applySecurity --source "/Projects" --principal "ICrod (LDAP)"
--permission "Read" --provider "ldap_LDAP" -z
```

Messaggi

Quando si utilizza questa API potrebbero essere visualizzati i seguenti messaggi:

- <permission> permission set successfully for <source>.
- <source> No such file or folder exists. Please try again.
- <permission> Invalid permission type, Please try again.
- <source> Error setting security ACL.

L'operazione cascadeSecurity

Estende le impostazioni di sicurezza della cartella a tutti i file e sottocartelle all'interno della cartella.

Sintassi

```
python CADSTool.py cascadeSecurity --source "<source>" -z
```

Il valore <source> è il percorso completo della cartella nel repository. È un parametro obbligatorio.

Esempio

```
python CADSTool.py cascadeSecurity --source "/Projects" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi durante l'utilizzo di questa API:

- Security ACL cascaded successfully for <source>.
- <source> No such folder exists. Please try again.
- <source> Error cascading security ACL.

L'operazione copyResource

Copia un file o cartella in un'altra cartella nel repository.

Per questa API è fornita una funzione di ridenominazione con cui è possibile rinominare il file specificato quando viene copiato. Anche ai casi descritti all'inizio di ["L'operazione moveResource" a pagina 17](#) si applica questa API copyResource.

Sintassi

```
python CADSTool.py copyResource --source "<source>" --target "<target>" -z
```

In cui:

- <source> è il percorso completo del repository di contenuti del file/cartella da copiare. È un parametro obbligatorio.
- <target> è il percorso di repository completo in cui il file/cartella deve essere copiato. È un parametro obbligatorio.

Esempi

L'esempio seguente copia un file:

```
python CADSTool.py copyResource --source "/Demo/Drafts/MyReport.rptdesign" --target  
"/Projects" -z
```

Il seguente esempio copia e rinomina un file:

```
python CADSTool.py copyResource --source "/Demo/Drafts/MyReport.rptdesign" --target  
"/Projects/Report.rptdesign" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi durante l'utilizzo di questa API:

- `<source>` copied successfully.
- `<source>` No such file or folder exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error copying file or folder.

L'operazione createFolder

Crea una nuova cartella in una posizione specificata nel repository.

Sintassi

```
python CADSTool.py createFolder --source "<source>" -z
```

Il valore `<source>` è il percorso completo della nuova cartella da creare. È un parametro obbligatorio. In base al percorso fornito, viene creata la nuova cartella, incluso le relative sottocartelle.

Esempio

Il seguente esempio crea *Bozze* se non esiste già.

```
python CADSTool.py createFolder --source "/Demo/Drafts" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi durante l'utilizzo di questa API:

- `<source>` Folder created successfully.
- `<source>` No such folder exists. Please try again.
- `<folder>` Folder already exists. Please try again.
- `<source>` Error creating folder.

L'operazione deleteFile

Elimina un file dal repository incluso tutte le relative versioni.

Sintassi

```
python CADSTool.py deleteFile --source "<source>" --submittedHierarchy -z
```

In cui:

- `<source>` è il percorso completo di repository del file da eliminare. È un parametro obbligatorio.
- `--submittedHierarchy` elimina un file dalla cartella Lavori inviati. È un parametro facoltativo.

Esempio

Il seguente esempio elimina un file dal repository, incluso tutte le relative versioni:

```
python CADSTool.py deleteFile --source "/Demo/Drafts/MyReport.rptdesign" -z
```

Il seguente esempio elimina un file dalla cartella Lavori inviati, incluso tutte le relative versioni:

```
python CADSTool.py deleteFile --source "Submitted Jobs/admin/2007-05-21.14.10.22.422-test.dbq/test.dbq.html" --submittedHierarchy -z
```

Messaggi

È possibile visualizzare i seguenti messaggi durante l'utilizzo di questa API:

- `<source>` deleted successfully.
- `<source>` No such file exists. Please try again.
- `<source>` Error deleting file.

L'operazione deleteFileVersion

Elimina una versione specifica di un file dal repository.

Sintassi

```
python CADSTool.py deleteFileVersion --source "<source>" --version "<version>"  
--label "<label>" --submittedHierarchy -z
```

In cui:

- `<source>` è il percorso completo di repository del file da eliminare. È un parametro obbligatorio.
- `<version>` è la versione specifica del file da eliminare. È un parametro facoltativo.
- `<label>` è l'etichetta del file da eliminare. È un parametro facoltativo.
- `--submittedHierarchy` elimina una specifica versione di un file dalla cartella Lavori inviati. È un parametro facoltativo.

Esempi

Il seguente esempio elimina una specifica versione di un file:

```
python CADSTool.py deleteFileVersion --source "/Demo/Drafts/MyReport.rptdesign" --version  
"0:2006-08-25 21:15:49.453" -z
```

Il seguente esempio elimina un file con una specifica etichetta:

```
python CADSTool.py deleteFileVersion --source "/Demo/Drafts/MyReport.rptdesign" --label  
"Test" -z
```

Il seguente esempio elimina un file con una specifica etichetta dalla cartella Lavori inviati:

```
python CADSTool.py deleteFileVersion --source "Submitted Jobs/admin/  
2007-05-21.14.10.22.422-test.dbq/test.dbq.html" --label "Test" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` deleted successfully.
- `<source>` No such file exists. Please try again.
- `<source>` Error deleting file.

Operazione deleteFolder

Elimina una cartella dal repository, incluso tutti i relativi contenuti.

Sintassi

```
python CADSTool.py deleteFolder --source <source> --submittedHierarchy -z
```

In cui:

- *<source>* è il percorso di repository completo della cartella da eliminare. È un parametro obbligatorio.
- `--submittedHierarchy` elimina una specifica versione della cartella dalla cartella Lavori inviati. È un parametro facoltativo.

Esempi

Il seguente esempio elimina una cartella:

```
python CADSTool.py deleteFolder --source "/Demo/Drafts" -z
```

Il seguente esempio elimina una cartella dalla cartella Lavori inviati:

```
python CADSTool.py deleteFolder --source "Submitted Jobs/admin/2007-05-21.14.10.22.422-test.dbq/" --submittedHierarchy -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- *<source>* deleted successfully.
- *<source>* No such folder exists. Please try again.
- *<source>* Error deleting folder.

L'operazione downloadFile

Scarica una versione specifica di un file dal repository nel file system locale.

Sintassi

```
python CADSTool.py downloadFile --source "<source>" --version "<version>" --label "<label>" --target "<target>" -z
```

In cui:

- *<source>* è il percorso di repository completo. È un parametro obbligatorio.
- *<version>* è la versione del file da scaricare. È un parametro facoltativo.
- *<label>* è l'etichetta del file da scaricare. È un parametro facoltativo.
- *<target>* è il percorso completo (sul file system locale) in cui il file deve essere scaricato..

Esempi

Il seguente esempio scarica l'ultima versione del file:

```
python CADSTool.py downloadFile --source "/Demo/Drafts/MyReport.rptdesign" --target "C:/Demo/Shared/" -z
```

Il seguente esempio scarica una versione specifica del file utilizzando un contrassegno di versione:

```
python CADSTool.py downloadFile --source "/Demo/Drafts/MyReport.rptdesign" --version "0:2006-08-25 21:15:49.453" --target "C:/Demo/Shared/" -z
```

Il seguente esempio scarica una versione con etichetta del file:

```
python CADSTool.py downloadFile --source "/Demo/Drafts/MyReport.rptdesign" --label "Production" --target "C:/Demo/Shared/" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` File downloaded successfully.
- `<source>` No such file exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error downloading File.

L'operazione di esportazione

Avvia un'esportazione dal Repository dei contenuti, consentendo di selezionare quali file e cartelle esportare e salvando il file di esportazione `*.pes` nel file system locale.

Sintassi

```
python CADSTool.py export --source "<source>" --target "<target>" -z
```

In cui:

- `<source>` è il percorso di repository completo della cartella da esportare. È un parametro obbligatorio.
- `<target>` è il percorso completo (sul file system locale) per il file di esportazione `*.pes` da creare. È un parametro obbligatorio.

Esempio

```
python CADSTool.py export --source "/Projects/" --target "C:\Demo\drafts.pes" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` exported successfully.
- `<source>` No such folder exists. Please try again.
- `<source>` Error exporting folder.

L'operazione `getAccessControlList`

Richiama l'ACL (access control list) di sicurezza per un file/cartella specificato nel repository dei contenuti.

Sintassi

```
python CADSTool.py getAccessControlList --source "<source>" -z
```

Il valore `<source>` è il percorso completo di file/cartella. È un parametro obbligatorio.

Esempio

```
python CADSTool.py getAccessControlList --source "/Projects/MyReport.rptdesign" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` No such file or folder exists. Please try again.
- Error retrieving security details for `<source>`.

L'operazione `getAllVersions`

Recupera un elenco di tutte le versioni di un file nel repository.

Sintassi

```
python CADSTool.py getAllVersions --source "<source>" --submittedHierarchy -z
```

In cui:

- `<source>` è il percorso repository completo del file di cui recuperare le versioni. È un parametro obbligatorio.
- `--submittedHierarchy` recupera le versioni dalla cartella Lavori inviati. È un parametro facoltativo.

Esempi

Il seguente esempio recupera tutte le versioni di un file specificato:

```
python CADSTool.py getAllVersions --source "/Demo/Drafts/MyReport.rptdesign" -z
```

Il seguente esempio recupera tutte le versioni di un file specificato dalla cartella Lavori inviati:

```
python CADSTool.py getAllVersions --source "Submitted Jobs/admin/2007-05-21.14.10.22.422-test.dbq/test.dbq.html" --submittedHierarchy -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file versions.
- Una volta completato correttamente il processo, vengono visualizzate le informazioni per ogni versione del file incluso le informazioni relative al contrassegno di versione e all'etichetta.

L'operazione getChildren

Recupera l'elenco di tutti i file e cartelle in una cartella specifica del repository.

Sintassi

```
python CADSTool.py getChildren --source "<source>" -z
```

Il valore di `<source>` è il percorso completo della cartella. È un parametro obbligatorio.

Esempio

```
python CADSTool.py getChildren --source "/Demo/Drafts" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Una volta completato il comando correttamente, elenca tutti i contenuti della cartella specificata.
- `<source>` No such folder exists. Please try again.
- `<source>` Error getting resources.

L'operazione getCustomPropertyValue

Recupera i valori validi accettati da una proprietà personalizzata specificata.

Sintassi

```
python CADSTool.py getCustomPropertyValue --propertyName "<propertyName>" -z
```

Il valore `<propertyName>` è il nome della proprietà personalizzata. È un parametro facoltativo.

Esempio

```
python CADSTool.py getCustomPropertyValue --propertyName "Language" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<propertyName>` takes values as `<valid values>`
- Error retrieving property details for `<propertyName>`.

L'operazione getMetadata

Recupera gli attributi di metadati di un file o cartella nel repository.

Sintassi

```
python CADSTool.py getMetadata --source "<source>" --version "<version>" --label  
"<label>" --submittedHierarchy -z
```

In cui:

- `<source>` è il percorso di repository completo del file o della cartella per richiamare i metadati. Per le cartelle, vengono ignorati gli attributi version/label. È un parametro obbligatorio.
- `<version>` è la versione del file per cui recuperare i metadati. È un parametro facoltativo.
- `<label>` è l'etichetta del file per cui recuperare i metadati. È un parametro facoltativo.
- `--submittedHierarchy` recupera i metadati dalla cartella Lavori inviati. È un parametro facoltativo.

Esempi

Il seguente esempio recupera i metadati per una cartella:

```
python CADSTool.py getMetadata --source "/Demo/Drafts" -z
```

Il seguente esempio recupera i metadati per una versione con etichetta di un file:

```
python CADSTool.py getMetadata --source "/Demo/Drafts/MyReport.iptdesign" --label "Test" -z
```

Il seguente esempio recupera i metadati per una versione con etichetta di un file nella cartella Lavori inviati:

```
python CADSTool.py getMetadata --source "Submitted Jobs/admin/  
2007-05-21.14.10.22.422-test.dbq/test.dbq.html" --label "LATEST" --submittedHierarchy -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file metadata.
- Una volta completato correttamente il processo, vengono visualizzate tutte le informazioni relative ai metadati per il file o cartella specificato incluso le proprietà personalizzate dei metadati.

L'operazione di importazione

Importa un file di esportazione *.pes esistente dal file system locale nel repository.

Sintassi

```
python CADSTool.py import --source "<source>" --target "<target>"  
--resourceType "<type>"  
--resourceConflict "<rconflict>"  
--labelFrom "<label>"  
--lockResolution "<resolution>"  
--invalidVersionConflict "<vconflict>"
```

```
--resourceDef "<rdefinition>"
--exclude "<exclusion>"
-z
```

In cui:

- `<source>` è il percorso completo (sul file system locale) del file di esportazione *.pes da importare nel repository. È un parametro obbligatorio.
- `<target>` è il percorso di repository completo in cui importare il file di esportazione *.pes. È un parametro obbligatorio.
- `<type>` indica il tipo di contenuto che si sta importando. Specificare uno dei seguenti valori:
 - **ContentRepository** per gli oggetti contenuto come file e cartelle.
 - **ResourceDef** per le definizioni di risorsa
 - **Credential** per le credenziali utente
 - **DataSource** per le definizioni origine dati
 - **MessageDomain** per i domini di messaggio
 - **ServerCluster** per le definizioni di cluster server
 - **Server** per le definizioni server
 - **PromotionPolicy** per le politiche di promozione

È un parametro facoltativo. Se non viene specificato questo parametro, viene utilizzato il valore predefinito **ContentRepository**.

- `<rconflict>` indica in che modo risolvere i conflitti di duplicazione di ID o nome. Specificare uno dei seguenti valori:
 - **keepTarget**. Viene conservato l'elemento di destinazione. L'elemento di origine con l'ID duplicato, contenuto nel file .pes, viene ignorato.
 - **addNewVersion**. Solitamente questa opzione si utilizza per risolvere i conflitti di ID o di nomi. Se tra l'oggetto di origine e quello di destinazione si verifica un conflitto a causa di un ID duplicato, nell'ubicazione di destinazione viene creata una nuova versione dell'oggetto. In caso di conflitto tra nomi, l'oggetto importato viene rinominato nell'ubicazione di destinazione. Generalmente, agli oggetti rinominati viene fatto seguire il suffisso _1, _2 e così via. Nel caso in cui due versioni di un oggetto abbiano la stessa etichetta, il sistema ne conserva una ed elimina il doppione perché due versioni dello stesso elemento non possono avere la stessa etichetta. L'etichetta conservata dipende dal parametro **labelFrom**.

È un parametro facoltativo. Se non viene specificato questo parametro, verrà utilizzato il valore predefinito **keepTarget**.

- `<label>` specifica quale etichetta utilizzare se due versioni di un oggetto hanno la stessa etichetta. L'etichetta per l'altra versione viene eliminata. Specificare uno dei seguenti valori:
 - **source**
 - **target**

È un parametro facoltativo. Se non viene specificato questo parametro, viene utilizzato il valore predefinito **source**.

- `<resolution>` definisce in che modo continuare se vengono rilevate risorse bloccate. Specificare uno dei seguenti valori:
 - **continue**. Continua l'importazione, omettendo le risorse bloccate.
 - **abort**. Termina il processo di importazione se si rilevano risorse bloccate. In caso di conflitti dovuti al blocco di oggetti, l'importazione viene interrotta e non va a buon fine.

È un parametro facoltativo. Se non viene specificato questo parametro, viene utilizzato il valore predefinito **abort**.

- `<vconflict>` definisce in che modo continuare se viene rilevata una versione non valida durante il processo di importazione. Specificare uno dei seguenti valori:

- **import**. Verrà importata la versione non valida.
- **discard**. Verrà eliminata la versione non valida.

È un parametro facoltativo. Se non viene specificato questo parametro, viene utilizzato il valore predefinito **import**.

- *<rdefinition>* definisce il comportamento di elaborazione delle definizioni delle risorse. Specificare uno dei seguenti valori:
 - **recommended**. Una definizione risorsa viene importata solo se l'identificativo o il nome non è in conflitto con una definizione di destinazione. Tutte le definizioni di risorse che presentano un conflitto non vengono importate.
 - **include**. Tutte le definizioni di risorsa dai file di importazione vengono importate. È possibile selezionare uno o più tipi di definizione risorsa da escludere dall'importazione selezionando la casella di spunta corrispondente.
 - **exclude**. Nessuna definizione di risorsa dai file di importazione viene importata. Gli oggetti importati possono essere modificati perché facciano riferimento a definizioni delle risorse disponibili.

È un parametro facoltativo. Se non viene specificato questo parametro, viene utilizzato il valore predefinito **recommended**.

- *<exclusion>* definisce quali tipi di risorsa sono esclusi durante l'importazione. È possibile combinare più valori in qualsiasi ordine come un elenco delimitato da punto e virgola. Specificare uno o più dei seguenti valori:
 - **credential** esclude le credenziali utente
 - **customproperty** esclude le proprietà personalizzate per gli oggetti risorsa.
 - **datasource** esclude le definizioni di origine dati
 - **messagedomain** esclude i domini del messaggio
 - **notification** esclude le definizioni di notifica
 - **servercluster** esclude le definizioni di cluster server
 - **server** esclude le definizioni di server
 - **topic** esclude le definizioni di argomento

È un parametro facoltativo. Se non viene specificato questo parametro, nell'importazione vengono inclusi tutti i tipi.

Esempio

```
python CADSTool.py import --source "C:\Demo\drafts.pes" --target "/Demo/Drafts/"
--resourceConflict "addNewVersion" --labelFrom "target" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- *<source>* imported successfully.
- *<source>* No such file exists. Please try again.
- *<target>* No such folder exists. Please try again.
- *<source>* Error importing folder.

L'operazione moveResource

Sposta un file o cartella in un'altra cartella nel repository.

Per questa API viene fornita una funzione di ridenominazione in cui è possibile rinominare un file/cartella quando viene spostato. I seguenti casi descrivono il comportamento della funzione di ridenominazione:

Se l'origine è */Temp Folder/Temp.txt* e la destinazione è */Cartella Demo* :

- **Caso 1** : se esiste la cartella *Cartella Demo*, *Temp.txt* viene spostata in *Cartella Demo*.

- **Caso 2:** Se non esiste la cartella *Cartella Demo*, *Temp.txt* viene spostata in "/" e rinominata nella *Cartella Demo*.

Se l'origine è */Temp Folder/Temp.txt* e la destinazione è */Demo Folder/Abc.dat*:

- **Caso 1:** se esiste la cartella *Cartella Demo*, *Temp.txt* viene spostata in *Cartella Demo* e rinominata in *Abc.dat*.
- **Caso 2:** se non esiste la cartella *Cartella Demo*, viene visualizzato un errore.

Sintassi

```
python CADSTool.py moveResource --source "<source>" --target "<target>" -z
```

In cui:

- *<source>* è il percorso di repository completo del file/cartella da spostare. È un parametro obbligatorio.
- *<target>* è il percorso di repository completo in cui il file/cartella deve essere spostato. È un parametro obbligatorio.

Esempi

Il seguente esempio mostra come spostare un file:

```
python CADSTool.py moveResource --source "/Demo/Drafts/MyReport.rptdesign" --target
"/Approved" -z
```

Il seguente esempio mostra come spostare una cartella:

```
python CADSTool.py moveResource --source "/Demo/Drafts/" --target "/Projects" -z
```

Il seguente esempio mostra come spostare e rinominare un file:

```
python CADSTool.py moveResource --source "/Demo/Drafts/MyReport.rptdesign" --target
"/Approved/Report.rptdesign" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- *<source>* moved successfully.
- *<source>* No such file or folder exists. Please try again.
- *<target>* No such folder exists. Please try again.
- *<source>* Error moving file or folder.

L'operazione removeLabel

Rimuove un'etichetta da un file nel repository.

Sintassi

```
python CADSTool.py removeLabel --source "<source>" --label "<label>" -z
```

In cui:

- *<source>* è il percorso completo del file nel repository. È un parametro obbligatorio.
- *<label>* è il nome dell'etichetta da rimuovere dal file specificato. È un parametro obbligatorio.

Esempio

```
python CADSTool.py removeLabel --source "/Demo/Drafts/MyReport.rptdesign"
--label "Draft" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Label removed successfully for <source>.
- <source> No such folder exists. Please try again.
- <source> Error deleting label.
- <label> No such label exists. Please try again.

L'operazione removeSecurity

Rimuove l'ACL (Access Control List) di sicurezza da un file o cartella specificati nel repository.

Sintassi

```
python CADSTool.py removeSecurity --source "<source>" --principal "<principal>"  
--provider "<provider>" --cascade -z
```

In cui:

- <source> è il percorso completo del file/cartella da cui rimuovere la sicurezza. È un parametro obbligatorio.
- <principal> è l'utente/principal (ad esempio *admin*) cui rimuovere la sicurezza per il file/cartella specificato. È un parametro obbligatorio.
- <provider> è il provider di sicurezza da utilizzare per il richiamo delle informazioni sugli utenti (principal). È un parametro facoltativo. I valori validi includono:
 - Native per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito.
 - AD_<name> per Active Directory, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema
 - ADL_<name> per Active Directory con sovrascrittura locale, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema
 - ldap_<name> per OpenLDAP, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema
- --cascade viene utilizzato durante la rimozione della sicurezza da una cartella, per rimuovere le impostazioni di sicurezza da tutti i file e sottocartelle all'interno della cartella specificata. È un parametro facoltativo.

Esempio

```
python CADSTool.py removeSecurity --source "/Projects/MyReport.rptdesign"  
--principal "icrod" --provider "Native" --cascade -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- <source> All the security ACL removed successfully.
- <source> No such folder exists. Please try again.
- <source> Error deleting security ACL.

L'operazione di ricerca

Ricerca i file e cartelle nel repository. I risultati generano un elenco di file/cartelle e relative versioni corrispondenti al criterio di ricerca.

Sintassi

```
python CADSTool.py search --criteria "<criteria>" -z
```

Il valore di `< criteria>` è la stringa di ricerca utilizzata per cercare metadati per tutti i file e le cartelle nel repository. È un parametro obbligatorio.

Esempio

```
python CADSTool.py search --criteria "Quarterly" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Una volta completata correttamente la ricerca, viene visualizzato un elenco di tutti i file e cartelle corrispondenti al criterio di ricerca. Questo elenco generalmente include i nomi file con i relativi percorsi completi e versioni.
- `<criteria> No file or folder matches the search criteria.`
- `Error searching files and folders.`

L'operazione setLabel

Applica un'etichetta ad una versione di un file nel repository. Se il file già dispone di un'etichetta, l'etichetta originale viene rimossa e sostituita dalla nuova.

Sintassi

```
python CADSTool.py setLabel --source "<source>" --version "<version>" --label "<label>" -z
```

In cui:

- `<source>` è il percorso completo del file nel repository. È un parametro obbligatorio.
- `<version>` è la versione del file a cui applicare l'etichetta. È un parametro obbligatorio.
- `<label>` è il nome dell'etichetta da applicare alla versione specificata del file. È un parametro obbligatorio.

Esempio

```
python CADSTool.py setLabel --source "/Demo/Drafts/MyReport.rptdesign" --version "1:2006-08-25 21:15:49.453" --label "Beta" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `Label set successfully for <source>.`
- `<source> No such folder exists. Please try again.`
- `<source> Error setting label.`

L'operazione setMetadata

Applica le proprietà dei metadati ai file e cartelle nel repository.

Tabella 4 a pagina 20 elenca le proprietà dei metadati e indica se possono essere applicate ai file e cartelle.

<i>Tabella 4. Proprietà dei metadati e tipi di risorsa</i>	
Proprietà metadati	Tipo di risorsa
Autore	File
Descrizione	File o cartella
Titolo	File o cartella

Tabella 4. Proprietà dei metadati e tipi di risorsa (Continua)

Proprietà metadati	Tipo di risorsa
Data di scadenza	File o cartella
Parola chiave	File
Argomenti	File
Metadati personalizzati	File o cartella

Sintassi

```
python CADSTool.py setMetadata --source "<source>" --version "<version>" --label
"<label>" --author "<author>" --title "<title>" --description "<description>"
--expirationDate "<expirationDate>" --topic "<topic>" --keyword "<keyword>"
--customProperty "<customProperty>" -z
```

In cui:

- `<source>` è il percorso di repository completo del file o della cartella su cui impostare i metadati. È un parametro obbligatorio.
- `<author>` è l'autore del file o della cartella. È un parametro facoltativo.
- `<title>` è il titolo del file o della cartella. È un parametro facoltativo.
- `<description>` è la descrizione del file/cartella. È un parametro facoltativo.
- `<expirationDate>` è la data di scadenza del file o della cartella. È un parametro facoltativo. Il formato data è YYYY-MM-DDThh:mm:ssTZD (ad esempio, 1997-07-16T19:20:30+01:00), in cui:

YYYY = quattro cifre per l'anno

MM = due cifre per il mese (01 è gennaio, ecc.)

DD = due cifre per il giorno del mese da 01 a 31)

hh = due cifre per l'ora da (00 a 23, no am/pm)

mm = two-digit minute (00 through 59)

ss = due cifre per i secondi da (00 a 59)

TZD = time zone designator (Z or +hh:mm or -hh:mm)

- `<keyword>` è la parola chiave per il file o cartella. È un parametro facoltativo.
- `<version>` è la versione specifica del file su cui applicare i metadati. È un parametro facoltativo.
- `<label>` è la versione etichettata del file su cui applicare i metadati. È un parametro facoltativo.
- `<topic>` è l'argomento da applicare al file o alla cartella. È un parametro facoltativo.
- `<customProperty>` sono i valori di proprietà personalizzati da applicare al file o alla cartella. È un parametro facoltativo. I valori sono specificati come `<customProperty>=<value>`. Per applicare più di una proprietà personalizzata, utilizzare il punto e virgola (;) come un separatore (`<customProperty>=<value>;<customProperty>=<value>`). Separare i valori di proprietà a selezione multipla con l'operatore | (`<customProperty>=opt1|opt2;<customProperty>=value`).

Nota: È necessario fornire almeno un parametro facoltativo per utilizzare l'API setMetadata.

Esempio

```
python CADSTool.py setMetadata --source "/Demo/Drafts/MyReport.rptdesign" --version
"0:2006-08-25T21:15:49+01:00" --keyword "Trimestrale"
--customProperty "multi=hi|hello|bye;Complexity Degree=Simple" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- <source> Metadata set successfully.
- <source> No such file or folder exists. Please try again.
- <source> Error setting metadata.

L'operazione uploadFile

Salva un file nel repository di contenuti dal file system locale con la possibilità di creare una nuova versione dei file se già esiste.

Sintassi

```
python CADSTool.py uploadFile --source "<source>" --target "<target>" --createVersion -z
```

In cui:

- <source> è il percorso completo (sul file system locale) del file da caricare. È un parametro obbligatorio.
- <target> è il percorso completo della cartella nel repository in cui il file deve essere caricato. È un parametro obbligatorio.
- --createVersion indica che il file specificato già esiste e deve essere creata una nuova versione. È un parametro facoltativo.

Esempi

Nel seguente esempio, la destinazione è un percorso completo per *Bozze*:

```
python CADSTool.py uploadFile --source "C:\Demo\MyReport.rptdesign"
--target "/Demo/Drafts" -z
```

Se *MyReport.rptdesign* già esiste nella cartella */Demo/Bozze*, utilizzare il parametro --createVersion:

```
python CADSTool.py uploadFile --source "C:\Demo\MyReport.rptdesign"
--target "/Demo/Drafts" --createVersion -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- <source> File uploaded successfully.
- <source> No such file exists. Please try again.
- <target> No such folder exists. Please try again.
- <source> Error Uploading File.

Funzioni di gestione dei processi

Lo script di gestione dei processi consente di utilizzare i lavori. Quest'area comprende le seguenti funzionalità:

- Esecuzione dei lavori
- Richiamo della cronologia del lavoro
- Richiamo dei dettagli del lavoro

Questa sezione illustra l'utilizzo della riga di comando Python per le funzioni di gestione dei processi. Ogni API contiene le informazioni dettagliate di sintassi, un esempio e i messaggi previsti.

Parole chiave

Tabella 5 a pagina 23 elenca le parole chiave supportate per le API Process Management. La seconda colonna elenca la versione facoltativa, abbreviata delle parole chiave fornite. La tabella elenca solo le parole chiave specifiche delle API Process Management. Per altre parole chiave che si applicano ad entrambe le API Process Management e le API repository, consultare [Tabella 2 a pagina 3](#) e [Tabella 3 a pagina 5](#).

Parola chiave	Versione breve facoltativa	Uso
--source	-s	Il lavoro origine incluso il percorso
--target	-t	Il percorso di destinazione
--notification	-j	Indica che il lavoro verrà eseguito con notifiche
--async	-m	Indica che il lavoro verrà eseguito in modo asincrono
--execId	-y	L'ID esecuzione per il lavoro
--jobStepName	-q	Il nome fase lavoro
--log		Indica che i log non devono essere eliminati. Se utilizzata insieme a --target, i log vengono memorizzati in una posizione specificata da --target. Altrimenti, i log vengono visualizzati in linea.

Operazioni di gestione dei processi

L'operazione deleteJobExecutions

Elimina gli oggetti di esecuzione del lavoro specificato.

Sintassi

```
python CADSTool.py deleteJobExecutions --execIds "<execIDs>" -z
```

Il valore `<execIDs>` è un elenco delimitato da spazio degli identificativi per le esecuzioni da eliminare. È un parametro obbligatorio.

Esempi

```
python CADSTool.py deleteJobExecutions --execIds  
"0a58c33d002ce9080000 010e0ccf7b01800e" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Execution Id not specified.

L'operazione executeJob

Esegue un lavoro in modo sincrono o asincrono in base ai parametri passati. Nel caso di un'esecuzione sincrona, non viene restituita l'API fino a quando non è completato il lavoro. Nel caso di un'esecuzione asincrona, l'API viene restituita dopo che il lavoro è avviato.

Sintassi

```
python CADSTool.py executeJob --source "<source>" --notification --async -z
```

In cui:

- `<source>` è il percorso completo del lavoro nel repository. È un parametro obbligatorio.
- `--notification` viene utilizzato per eseguire il lavoro con notifiche. È un parametro facoltativo.
- `--async` viene utilizzato per eseguire il lavoro in modo asincrono. È un parametro facoltativo.

Esempi

Il seguente esempio esegue il lavoro in modo sincrono senza notifiche.

```
python CADSTool.py executeJob --source "/Demo/Jobs/Reports" -z
```

Il seguente esempio esegue il lavoro in modo sincrono con notifiche:

```
python CADSTool.py executeJob --source "/Demo/Jobs/Reports" --notification -z
```

Il seguente esempio esegue il lavoro in modo asincrono senza notifiche:

```
python CADSTool.py executeJob --source "/Demo/Jobs/Reports" --async -z
```

Il seguente esempio esegue il lavoro in modo asincrono con notifiche:

```
python CADSTool.py executeJob --source "/Demo/Jobs/Reports" --async --notification -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- `<source> Job executed successfully. Job execution Id is <execId>.`
- `<source> No such job exists. Please try again.`
- `<source> Error executing job.`

Operazione `getJobExecutionDetails`

Elenca i dettagli di esecuzione per un lavoro specifico, incluso le fasi lavoro e le iterazioni.

Sintassi

```
python CADSTool.py getJobExecutionDetails --execId "<execID>" --log --target  
"<target>" -z
```

In cui:

- `<execId>` è l'identificativo di esecuzione del lavoro. È un parametro obbligatorio.
- `--log` indica che il log del lavoro deve essere visualizzato in linea. Se non è incluso il parametro `--log`, qualsiasi log generato da una fase lavoro non viene visualizzato. È un parametro facoltativo.
- `<target>` è la posizione (sul file system locale) per memorizzare i log. Questo è un parametro facoltativo e può essere utilizzato solo insieme al parametro `--log`.

Esempi

Il seguente esempio elenca i dettagli di esecuzione di un lavoro specifico:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87  
b48400" -z
```

Il seguente esempio elenca i dettagli di esecuzione di un lavoro specifico con il log visualizzato in linea:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87
b48400" --log -z
```

Il seguente esempio elenca i dettagli di esecuzione di un lavoro specifico, con i log lavori memorizzati in una specifica posizione:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87
b48400" --log --target "c:\logs" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Per un'esecuzione corretta, vengono elencati tutti i dettagli per il lavoro, le fasi lavoro e le iterazioni lavoro. I log vengono visualizzati in linea o salvati in una posizione specifica sul file system locale.
- `<execId> No such execution exists. Please try again.`
- `<execId> Error displaying details of a job execution.`
- `--target cannot be used without --log parameter`

L'operazione `getJobExecutionList`

Elenca le esecuzioni correnti e le esecuzioni completate per uno lavoro specifico per tutte le versioni del lavoro.

Sintassi

```
python CADSTool.py getJobExecutionList --source "<source>" -z
```

Il valore `<source>` è il percorso completo del lavoro nel repository. È un parametro obbligatorio.

Esempio

```
python CADSTool.py getJobExecutionList --source "/Demo/Jobs/Reports" -z
```

Messaggi

È possibile visualizzare i seguenti messaggi quando si utilizza questa API:

- Per un'esecuzione corretta del lavoro specificato, vengono elencati tutti i dettagli dell'esecuzione come l'ID esecuzione, il nome lavoro, lo stato di esecuzione del lavoro e l'ora di inizio e di fine di esecuzione del lavoro.
- `<source> No such job exists. Please try again.`
- `<source> Error displaying execution list for a job.`

Capitolo 3. Il modulo PESImpl

La funzione IBM SPSS Collaboration and Deployment Services - Essentials for Python consente l'interazione con gli oggetti IBM SPSS Collaboration and Deployment Services Repository direttamente all'interno degli script Python.

All'interno del codice Python, importare la classe PESImpl dal modulo `pes.api.PESImpl`. Creare un oggetto PESImpl che utilizza le informazioni di connessione per il repository a cui effettuare la connessione.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("user", "password",
                 "host", "port",
                 ssl=True)
```

Di seguito sono riportati i parametri per il costruttore PESImpl:

- *user* corrisponde al nome dell'utente. Il valore deve includere un prefisso che indica il provider di sicurezza se l'utente non proviene dal provider nativo. Sono validi i valori di prefisso riportati di seguito:
 - Native per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito.
 - AD_<name> per Active Directory, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema
 - ADL_<name> per Active Directory con sovrascrittura locale, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema
 - ldap_<name> per OpenLDAP, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema

Inserire una barra ed il nome utente dopo il prefisso. Per i provider Active Directory, includere il dominio nel prefisso. Ad esempio, per l'utente *icrod* nel dominio *MYDOMAIN* dell'istanza Active Directory *AD_SPSSAD*, il valore è *AD_SPSSAD/MYDOMAIN/icrod*. Se l'utente *icrod* si trova nel provider OpenLDAP *SPSSLDAP*, il valore è *ldap_SPSSLDAP/icrod*.

- *password* corrisponde alla password associata all'utente specificato
- *host* indica il nome del server del repository
- *port* specifica il numero di porta per il server del repository
- *ssl=True* indica che il server del repository utilizza il protocollo SSL (secure sockets layer) per la codifica delle comunicazioni. Se il parametro *ssl* viene impostato su *False* o se il parametro viene omesso durante la creazione dell'oggetto PESImpl, le comunicazioni del server non utilizzano SSL. Quando si utilizza SSL, il server del repository deve essere configurato per SSL. Per ulteriori informazioni, consultare la documentazione dell'amministratore.

In alternativa, è possibile specificare l'URL del server invece dei parametri *host*, *port* e *ssl*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("user", "password",
                 server_url="url")
```

Il parametro *server_url* specifica l'URL completo per il server del repository. Utilizzare questo parametro se il server utilizza un percorso di contesto personalizzato. Ad esempio, il seguente costruttore corrisponde ad un server denominato `myserver` che utilizza SSL sulla porta 443 ed ha il percorso di contesto `/ibm/spss`:

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("myUser", "myPass",
                 server_url="https://myserver:443/ibm/spss")
```

Nota: Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre, come, ad esempio, `[3ffe:2a00:100:7031::1]`.

È quindi possibile accedere ai metodi specifici utilizzando l'oggetto `pesImpl`.

API repository di contenuti

Lo script del repository dei contenuti consente di utilizzare le risorse del repository, quali ad esempio file e cartelle. Quest'area comprende le seguenti funzionalità:

- Creazione ed eliminazione di cartelle
- Caricamento e scaricamento dei file
- Esportazione ed importazione delle cartelle
- Gestione delle etichette, della sicurezza e dei metadati

Questa sezione illustra l'API PESImpl utilizzata per gestire le risorse archiviate nel repository. Ogni metodo contiene le informazioni di sintassi dettagliate, un esempio e i messaggi previsti.

Metodi

Le seguenti sezioni elencano tutti i metodi di repository di contenuti supportati per IBM SPSS Collaboration and Deployment Services.

Nota:

- Per tutti i metodi con parametri facoltativi `Label` e `Version`, utilizzare `Label` o `Version`, ma non entrambi. Se, per un determinato file, non viene specificato il parametro `Version` o `Label`, viene utilizzata la versione più recente.
- Per tutti i metodi che richiedono un percorso per i file o cartelle nel repository, possono essere utilizzati il percorso o l'URI oggetto. L'URI oggetto può essere desunto visualizzando le proprietà dell'oggetto in IBM SPSS Deployment Manager.
- Per i metodi che richiedono l'input del repository di origine o di destinazione, o i percorsi del file system che contengono caratteri non Latin Unicode, le stringhe devono essere specificate come oggetti Unicode, ad esempio:

```
identificationSpecifier = pesImpl.uploadFile  
(source=u'C:\Analytics\La Peña.txt',  
target=u'/La Peña')
```

Il metodo `advanceSearch`

Ricerca i file e cartelle nel repository in base a vari parametri passati come input.

È possibile ricercare i seguenti elementi:

- Autore
- Descrizione
- Titolo
- Creato da
- Modificato da
- Data di inizio scadenza
- Data di fine scadenza
- Tipo MIME
- Etichetta
- Parola chiave
- Argomenti
- Data di inizio creazione
- Data di fine creazione
- Data di inizio versione modificata
- Data di fine oggetto modificato

- Data di inizio oggetto modificato
- Data di inizio versione modificata
- Data di fine versione modificata
- URI cartella padre
- URI risorsa

```
advanceSearch(criteriaDict,submittedHierarchy)
```

Tabella 6. Parametri di input per *advanceSearch*

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>criteriaDict</i>	Obbligatorio	Dizionario	<p>Il dizionario contiene la coppia chiave/valore rispetto alla ricerca che verrà effettuata. I valori chiave accettabili sono:</p> <ul style="list-style-type: none"> • author • title • description • createdBy • modifiedBy • expirationStartDate • expirationEndDate • mimeType • label • keyword • topic • createdStartDate • createdEndDate • objectModifiedStartDate • objectModifiedEndDate • versionModifiedStartDate • versionModifiedEndDate • parentURI • uri 	<pre>{ "author": "admin" , "title": "search" , "label": "label 1"} </pre>

Tabella 6. Parametri di input per `advanceSearch` (Continua)

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>submittedHierarchy</code>	Facoltativo	Booleano	Indica se ricercare la cartella <i>Lavori inviati</i>	True o False

Considerare che attualmente `expirationStartDate` e `expirationEndDate` non funzionano quando utilizzati in combinazione con altri campi di ricerca (come ad esempio titolo o autore).

Tabella 7. Valore di ritorno per `advanceSearch`

Tipo	Descrizione
<code>PageResult</code>	Una struttura in cui ciascuna riga corrisponde ad una corrispondenza nella ricerca. Vedere l'argomento "La classe PageResult" a pagina 58 per ulteriori informazioni.

Tabella 8. Eccezioni per `advanceSearch`

Tipo	Descrizione
<code>InsufficientParameterException</code>	Non sono specificati parametri obbligatori.

Esempio: ricerca per etichetta e parola chiave

Il seguente esempio restituisce tutte le versioni con etichetta *Produzione* che ha un valore di parola chiave *Trimestrale*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
critDict = {'label': 'Production', 'keyword': 'Quarterly'}
sResults = pesImpl.advanceSearch(critDict)
sRows = sResults.getRows()
for sRow in sRows:
    print "Author: ", sRow.getAuthor()
    print "Title: ", sRow.getTitle()
    for child in sRow.getChildRow():
        print "Version: ", child.getVersionMarker()
        print "Label: ", child.getVersionLabel()
        print "Keywords: ", child.getKeyword()
        print "URI: ", child.getUri()
```

Esempio: ricerca per URI

Il seguente esempio restituisce tutte le versioni file del file con l'URI specificato.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
critDict = {'uri': 'spsscr:///?id=a010a37ba5992bb00000127b0f952f945be'}
sResults = pesImpl.advanceSearch(critDict)
sRows = sResults.getRows()
for sRow in sRows:
    print "Author: ", sRow.getAuthor()
    print "Title: ", sRow.getTitle()
    for child in sRow.getChildRow():
        print "Version: ", child.getVersionMarker()
        print "Label: ", child.getVersionLabel()
        print "Keywords: ", child.getKeyword()
        print "URI: ", child.getUri()
```

Il metodo `applySecurity`

Imposta l'ACL di sicurezza per un file o cartella nel repository.

```
applySecurity(source, principal, permission, provider, cascade)
```

Tabella 9. Parametri di input per applySecurity

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	String	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	"/Temp Folder/ Temp.txt" oppure "0a58c3670016a7860000 010dcee0eaa28219"
<i>principal</i>	Obbligatorio	String	L'utente (come ad esempio <i>admin</i>) da utilizzare per il file o cartella specificato come parte di ACL	admin
<i>permission</i>	Obbligatorio	String	Il tipo di autorizzazione da applicare al file o cartella specificato.	READ, WRITE, DELETE, MODIFY_ACL 0 OWNER
<i>provider</i>	Facoltativo	String	Il provider di sicurezza da utilizzare per applicare la sicurezza agli utenti come ad esempio <i>Nativo</i>). I valori validi includono: <ul style="list-style-type: none"> • Native per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito. • AD_<name> per Active Directory, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema • ADL_<name> per Active Directory con sovrascrittura locale, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema • ldap_<name> per OpenLDAP, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema 	Native
<i>cascade</i>	Facoltativo	Booleano	Propaga le impostazioni di sicurezza a tutti i file e le cartelle presenti nella cartella specificata.	True o False

Tabella 10. Valore di ritorno per `applySecurity`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 11. Eccezioni per `applySecurity`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.
<code>IllegalParameterException</code>	L'utente o il nome del provider di sicurezza specificato non è corretto.

Esempio

Il seguente esempio mostra come assegnare l'autorizzazione `READ` per il file destinato ad un utente.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.applySecurity(source="/Projects", principal="icrod", permission="READ",
                                provider="Native")
```

Il metodo `cascadeSecurity`

Propaga le impostazioni di sicurezza della cartella a tutti i file e sottocartelle all'interno della cartella.

```
cascadeSecurity(source)
```

Tabella 12. Parametri di input per `cascadeSecurity`

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	String	Il percorso completo o l'URI dell'oggetto della cartella nel repository	"/Temp Folder" oppure "0a58c3670016a7860000010dcee0eaa28219"

Tabella 13. Valore di ritorno per `cascadeSecurity`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 14. Eccezioni per `ascadeSecurity`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	La cartella origine non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio mostra come applicare a cascata la sicurezza della cartella `Progetti` ai contenuti della cartella.

```
from pes.api.PESImpl import PESImpl
```

```
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.cascadeSecurity(source="/Projects")
```

Il metodo copyResource

Copia un file o cartella in un'altra cartella nel repository. Il file origine specificato o la cartella possono essere rinominati quando sono copiati.

Consultare [“Metodo moveResource”](#) a pagina 49 per ulteriori informazioni relative alla ridenominazione.

```
copyResource(source, target)
```

Tabella 15. Parametri di input per copyResource

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	String	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	"/Temp Folder/Temp.txt" oppure "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	Obbligatorio	String	Il percorso completo o l'URI oggetto della cartella in cui copiare il file. È anche possibile fornire un nuovo nome file per rinominare il file o cartella specificato quando viene copiato.	"/New Folder" oppure "/New Folder/abc.dat"

Tabella 16. Valore di ritorno per copyResource

Tipo	Descrizione
String	URI del file o cartella copiato

Tabella 17. Eccezioni per copyResource

Tipo	Descrizione
ResourceNotFoundException	Il file di origine o la cartella di destinazione non esiste.
InsufficientParameterException	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio copia la cartella *Bozze* in una cartella denominata *Progetti*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
uri = pesImpl.copyResource(source="/Demo/Drafts/MyReport.rptdesign", target="/Projects")
print uri
```

Il metodo createFolder

Crea una nuova cartella nella posizione specificata nel repository.

```
createFolder(source)
```

Tabella 18. Parametri di input per createFolder

Campo	Uso	Tipo	Descrizione	Valore di esempio
source	Obbligatorio	String	La cartella da creare nel repository	/Temp Folder/Sample Folder

Tabella 19. Valore di ritorno per createFolder

Tipo	Descrizione
String	URI della cartella creata

Tabella 20. Eccezioni per createFolder

Tipo	Descrizione
InsufficientParameterException	I parametri obbligatori non sono necessari.
ResourceAlreadyExistsException	La cartella specificata già esiste nel repository.

Esempio

Il seguente esempio mostra come creare una cartella denominata *Bozze* come figlio della cartella *Demo*. Se si verifica un problema durante la creazione della cartella, un messaggio di eccezione viene inviato alla console.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
try:
    uri = pesImpl.createFolder(source="/Demo/Drafts")
    print "URI for the folder is:", uri
except:
    print "Unhandled exception in createFolder."
```

Il metodo deleteFile

Elimina un file dal repository. Tutte le versioni del file vengono eliminate.

```
deleteFile(source, submittedHierarchy)
```

Tabella 21. Parametri di input per deleteFile

Campo	Uso	Tipo	Descrizione	Valore di esempio
source	Obbligatorio	String	Il percorso completo o l'URI dell'oggetto del file nel repository	"/Temp Folder/ Temp.txt" oppure "0a58c3670016a786 0000 010dcee0eaa28219"
submittedHierarchy	Facoltativo	Booleano	Indica se il file si trova nella cartella <i>Lavori inviati</i>	True o False

Tabella 22. Valore di ritorno per deleteFile

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 23. Eccezioni per deleteFile

Tipo	Descrizione
ResourceNotFoundException	Il file di origine non esiste.
InsufficientParameterException	I parametri obbligatori non sono necessari.
IllegalParameterException	La risorsa specificata da eliminare è una cartella.

Esempio

Il seguente esempio mostra come eliminare il file *MyReport.rptdesign* dal repository.

```

from pes.util.PESEExceptions import *
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
try:
    bSuccess = pesImpl.deleteFile(source="/Demo/Drafts/MyReport.rptdesign")
except ResourceNotFoundException:
    print "Specified file does not exist."
except InsufficientParameterException:
    print "No file specified."
except IllegalParameterException:
    print "Item to be deleted is not a file."
    
```

Il metodo deleteFileVersion

Elimina una specifica versione di un file dal repository.

```
deleteFileVersion(source, version, label, submittedHierarchy)
```

Tabella 24. Parametri di input per deleteFileVersion

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI oggetto del file nel repository	"/Temp Folder/ Temp.txt" o "0a58c3670016a78600 00010dcee0eaa28219"
<i>version</i>	Facoltativo. Tuttavia, è necessario specificare sia <i>version</i> che <i>label</i> .	Stringa	La versione specifica del file da eliminare	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facoltativo. Tuttavia, è necessario specificare sia <i>version</i> che <i>label</i> .	Stringa	La versione con etichetta specifica del file da eliminare.	"Versione 1"

Tabella 24. Parametri di input per deleteFileVersion (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
submittedHierarchy	Facoltativo	Booleano	Indica se il file è contenuto nella cartella <i>Lavori inviati</i>	True o False

Tabella 25. Valore di ritorno per deleteFileVersion

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 26. Eccezioni per deleteFileVersion

Tipo	Descrizione
ResourceNotFoundException	Il file origine o la cartella di destinazione non esiste.
InsufficientParameterException	I parametri obbligatori non sono specificati.
IllegalParameterException	La risorsa specificata da eliminare è una cartella.

Esempio

Il seguente esempio mostra come eliminare la versione del file *MyReport.rptdesign* con etichetta *Test* dal repository.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.deleteFileVersion(source="/Demo/Drafts/MyReport.rptdesign", label="Test")
```

Il metodo deleteFolder

Elimina una cartella e il relativo contenuto dal repository.

```
deleteFolder(source, submittedHierarchy)
```

Tabella 27. Parametri di input per deleteFolder

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
source	Obbligatorio	Stringa	Il percorso completo o l'URI oggetto della cartella nel repository	"/Temp Folder" o "0a58c3670016a7860000010dcee0eaa28219"
submittedHierarchy	Facoltativo	Booleano	Indica se la cartella è presente nella cartella <i>Lavori inviati</i>	True o False

Tabella 28. Valore di ritorno per deleteFolder

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 29. Eccezioni per deleteFolder

Tipo	Descrizione
ResourceNotFoundException	La cartella specificata non esiste.
InsufficientParameterException	I parametri obbligatori non sono specificati.
IllegalParameterException	La risorsa specificata da eliminare non è una cartella.

Esempio

Il seguente esempio mostra come eliminare la cartella denominata *Bozze* dal repository. Se si verifica un problema di eliminazione della cartella, viene inviato un messaggio di eccezione alla console.

```
from pes.util.PESEExceptions import *
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
try:
    bSuccess = pesImpl.deleteFolder(source="/Demo/Drafts")
except ResourceNotFoundException:
    print "Specified folder does not exist."
except InsufficientParameterException:
    print "No folder specified."
except IllegalParameterException:
    print "Item to be deleted is not a folder."
```

Il metodo downloadFile

Scarica una specifica versione di un file dal repository nel file system locale.

```
downloadFile(source, target, version, label)
```

Tabella 30. Parametri di input per downloadFile

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso repository completo del file da scaricare	"/Temp Folder/Temp.txt"
<i>target</i>	Obbligatorio	Stringa	Il percorso completo (sul file system locale) della cartella in cui scaricare il file.	"C:\Temp"
<i>version</i>	Facoltativo. Può essere specificato versione o etichetta.	Stringa	La versione specifica del file da scaricare.	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facoltativo. Può essere specificato versione o etichetta.	Stringa	La versione con etichetta specifica del file da scaricare.	"Version 2"

Tabella 31. Valore di ritorno per downloadFile

Tipo	Descrizione
Resource	Contenitore per le informazioni relative ad un oggetto repository. Consultare l'argomento "La classe Resource" a pagina 56 per ulteriori informazioni .

Tabella 32. Eccezioni per `downloadFile`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file origine o la cartella di destinazione non esiste.
<code>InsufficientParameterException</code>	Non sono specificati parametri obbligatori.

Esempio

Il seguente esempio mostra come scaricare una versione con etichetta *Produzione* del file *MyReport.rptdesign* per la directory *Condivisa* sul file system locale.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
resource = pesImpl.downloadFile(source="/Demo/Drafts/MyReport.rptdesign",
                                target="c:/Demo/Shared",label="Production")
```

Metodo `exportResource`

Esporta una cartella specifica del repository in un file di esportazione `*.pes` designato nel file system locale.

```
exportResource(source, target)
```

Tabella 33. Parametri di input di `exportResource`

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	Stringa	Il percorso completo del repository o l'URI dell'oggetto della cartella da esportare	/Temp Folder o 0a58c3670016a78 60000010dcee0eaa28219
<code>target</code>	Obbligatorio	Stringa	Il percorso completo (sul file system locale) ed il nome del file in cui esportare la cartella	C:\Temp\backup.pes

Tabella 34. Valore restituito di `exportResource`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 35. Eccezioni di `exportResource`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine o la cartella di destinazione non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.
<code>IllegalParameterException</code>	La destinazione specificata è una cartella. La destinazione deve essere un file <code>*.pes</code> .

Esempio

Il seguente esempio esporta i contenuti della cartella *Drafts* in un file nella cartella *backups* del file system locale.

```
from pes.api.PESImpl import PESImpl
```

```
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.exportResource(source="/Projects",target="C:\Demo\drafts.pes")
```

Metodo `getAccessControllist`

Recupera l'ACL (access control list) di sicurezza di un file o una cartella nel repository.

```
getAccessControllist(source,submittedHierarchy)
```

Tabella 36. Parametri di input di <code>getAccessControllist</code>				
Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	"/Temp Folder/ Temp.txt" o "0a58c3670016a78600 00010dcee0eaa28219"
<code>submittedHierarchy</code>	Facoltativo	Booleano	Indica se il file o la cartella si trova nella cartella <i>Lavori inviati</i>	True o False

Tabella 37. Valore restituito di <code>getAccessControllist</code>	
Tipo	Descrizione
Dictionary	Viene visualizzato un dizionario con i nomi utente e le autorizzazioni associate. Ad esempio: <pre>{ "admin": "MODIFY_ACL", "Joe": "DELETE" }</pre>

Tabella 38. Eccezioni di <code>getAccessControllist</code>	
Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine o la cartella di destinazione non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio stampa l'ACL del file `MyReport.rptdesign`.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
aclDic = pesImpl.getAccessControllist(source = "/Projects/MyReport.rptdesign")
print aclDic
```

Metodo `getAllVersions`

Recupera un elenco di tutte le versioni di un file nel repository.

```
getAllVersions(source,submittedHierarchy)
```

Tabella 39. Parametri di input di *getAllVersions*

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto del file nel repository	"/Temp Folder/ Temp.txt" o "0a58c3670016a78600 00010dcee0eaa28219"
<i>submittedHierarchy</i>	Facoltativo	Booleano	Indica se il file si trova nella cartella <i>Lavori inviati</i>	True o False

Tabella 40. Valore restituito di *getAllVersions*

Tipo	Descrizione
List	Un elenco degli oggetti della risorsa. Vedere "La classe Resource" a pagina 56.

Tabella 41. Eccezioni di *getAllVersions*

Tipo	Descrizione
ResourceNotFoundException	Il file di origine non esiste.
InsufficientParameterException	I parametri obbligatori non sono necessari.
IllegalParameterException	L'origine specificata è una cartella.

Esempio

Questo esempio recupera le informazioni sulle versioni del file *MyReport.rptdesign*, stampando l'autore, il contrassegno di versione e le etichette della versione di ognuno.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
resourceList = pesImpl.getAllVersions(source="/Demo/Drafts/MyReport.rptdesign")
for resource in resourceList:
    print resource.getAuthor()
    print resource.getVersionMarker()
    print resource.getVersionLabel()
```

Metodo *getChildren*

Recupera un elenco di tutti i file e cartelle presenti in una cartella specificata del repository.

```
getChildren(source, submittedHierarchy)
```

Tabella 42. Parametri di input di *getChildren*

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto della cartella nel repository	"/Temp Folder" o "0a58c3670016a78 600 00010dcee0eaa282 19"

Tabella 42. Parametri di input di `getChildren` (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>submittedHierarchy</code>	Facoltativo	Boolean	Indica se la cartella si trova nella cartella Lavori inviati	True o False

Tabella 43. Valore restituito di `getChildren`

Tipo	Descrizione
List	Un elenco degli oggetti della risorsa. Vedere “La classe Resource” a pagina 56.

Tabella 44. Eccezioni di `getChildren`

Tipo	Descrizione
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.
<code>ResourceNotFoundException</code>	La cartella specificata non esiste.

Esempio

Il seguente esempio recupera i contenuti della cartella `/Demo/Drafts`, stampando il titolo, l'autore e l'identificativo della risorsa di ciascuno.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
resourceList = pesImpl.getChildren(source="/Demo/Drafts")
for resource in resourceList:
    print "Resource title:", resource.getTitle()
    print "Resource author:", resource.getAuthor()
    print "Resource ID:", resource.getResourceID()
```

Metodo `getCustomPropertyValue`

Recupera i valori validi accettati da una proprietà personalizzata specificata.

```
getCustomPropertyValue(propertyName)
```

Tabella 45. Parametri di input di `getCustomPropertyValue`

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>propertyName</code>	Obbligatorio	Stringa	Il nome della proprietà personalizzata	"FreeForm"

Tabella 46. Valore restituito di `getCustomPropertyValue`

Tipo	Descrizione
List	Restituisce un elenco di valori validi accettati dalla proprietà personalizzata. Se la proprietà richiede una selezione (ad esempio, una selezione singola o multipla), l'elenco contiene tutti i valori validi per la selezione. Se si tratta di una proprietà free-form, l'elenco contiene il tipo di dati accettato dalla proprietà (ad esempio, String, Date, o Number).

Tabella 47. Eccezioni di `getCustomPropertyValue`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	La proprietà specificata non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio accede ai valori della proprietà personalizzata *Language*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
langList = pesImpl.getCustomPropertyValue(propertyName = "Language")
print langList
```

Metodo getMetadata

Recupera gli attributi dei metadati di un file o di una cartella nel repository, comprese qualsiasi proprietà personalizzata e le informazioni sull'argomento.

```
getMetadata(source, version, label, submittedHierarchy)
```

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	"/Temp Folder/ Temp.txt" o "0a58c3670016a7860 0 00010dcee0eaa28219 "
<i>version</i>	Facoltativo. È possibile specificare version o label.	Stringa	La versione specifica del file o della cartella	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facoltativo. È possibile specificare version o label.	Stringa	La versione etichettata specifica del file o della cartella	"Version 1"
<i>submittedHierarchy</i>	Facoltativo	Booleano	Indica se il file si trova nella cartella <i>Lavori inviati</i>	True o False

Tipo	Descrizione
Resource	Contenitore delle informazioni sull'oggetto del repository. Per ulteriori informazioni, consultare l'argomento "La classe Resource" a pagina 56.

Tipo	Descrizione
ResourceNotFoundException	Il file di origine o la cartella non esiste.
InsufficientParameterException	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio accede l'identificativo della risorsa della cartella */Demo/Drafts*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
resource = pesImpl.getMetadata(source="/Demo/Drafts")
resourceid = resource.getResourceID()
```

Metodo importResource

Importa un file di esportazione *.pes esistente dal file system locale nel repository.

```
importResource(source, target, resourceType, resourceConflict, invalidVersionConflict,
resourceDef, labelFrom, lockResolution, exclude)
```

Tabella 51. Parametri di input di importResource

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo (sul file system locale) del file da importare	"C:\Temp\New.pes"
<i>target</i>	Obbligatorio	Stringa	Il percorso completo del repository o l'URI dell'oggetto della cartella da esportare	"/Temp Folder" o "0a58c3670016a786000010dcee0eaa28219"
<i>resourceType</i>	Facoltativo	Stringa	Tipo di contenuto da importare. Specificare uno dei seguenti valori: <ul style="list-style-type: none">• ContentRepository per gli oggetti contenuto come file e cartelle.• ResourceDef per le definizioni di risorsa• Credential per le credenziali utente• DataSource per le definizioni origine dati• MessageDomain per i domini di messaggio• ServerCluster per le definizioni di cluster server• Server per le definizioni server• PromotionPolicy per le politiche di promozione Se non viene specificato questo parametro, viene utilizzato il valore predefinito ContentRepository .	

Tabella 51. Parametri di input di `importResource` (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>resourceConflict</code>	Facoltativo	Stringa	<p>Indica come risolvere l'ID duplicato o i nomi dei conflitti. Specificare uno dei seguenti valori:</p> <ul style="list-style-type: none"> • keepTarget. Viene conservato l'elemento di destinazione. L'elemento di origine con l'ID duplicato, contenuto nel file <code>.pes</code>, viene ignorato. • addNewVersion. Solitamente questa opzione si utilizza per risolvere i conflitti di ID o di nomi. Se tra l'oggetto di origine e quello di destinazione si verifica un conflitto a causa di un ID duplicato, nell'ubicazione di destinazione viene creata una nuova versione dell'oggetto. In caso di conflitto tra nomi, l'oggetto importato viene rinominato nell'ubicazione di destinazione. Generalmente, agli oggetti rinominati viene fatto seguire il suffisso <code>_1</code>, <code>_2</code> e così via. Nel caso in cui due versioni di un oggetto abbiano la stessa etichetta, il sistema ne conserva una ed elimina il doppio perché due versioni dello stesso elemento non possono avere la stessa etichetta. L'etichetta conservata dipende dal parametro labelFrom. <p>Se non viene specificato questo parametro, verrà utilizzato il valore predefinito keepTarget.</p>	
<code>labelFrom</code>	Facoltativo	Stringa	<p>Etichetta da utilizzare se due versioni di un oggetto hanno la stessa etichetta. L'etichetta dell'altra versione viene eliminata. Specificare source o target. Se non viene specificato questo parametro, viene utilizzato il valore predefinito source.</p>	

Tabella 51. Parametri di input di `importResource` (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>lockResolution</code>	Facoltativo	Stringa	<p>Specifica come procedere nel caso vi siano risorse bloccate. Specificare uno dei seguenti valori:</p> <ul style="list-style-type: none"> • continue. Continua l'importazione, omettendo le risorse bloccate. • abort. Termina il processo di importazione se si rilevano risorse bloccate. In caso di conflitti dovuti al blocco di oggetti, l'importazione viene interrotta e non va a buon fine. <p>Se non viene specificato questo parametro, viene utilizzato il valore predefinito abort.</p>	
<code>invalidVersionConflict</code>	Facoltativo	Stringa	<p>Definisce come procedere nel caso in cui si incontra una versione non valida nel processo di importazione. Specificare uno dei seguenti valori:</p> <ul style="list-style-type: none"> • import. Verrà importata la versione non valida. • discard. Verrà eliminata la versione non valida. <p>Se non viene specificato questo parametro, viene utilizzato il valore predefinito import.</p>	

Tabella 51. Parametri di input di importResource (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>resourceDef</i>	Facoltativo	Stringa	<p>Definisce il comportamento di elaborazione delle definizioni delle risorse. Specificare uno dei seguenti valori:</p> <ul style="list-style-type: none"> • recommended. Una definizione risorsa viene importata solo se l'identificativo o il nome non è in conflitto con una definizione di destinazione. Tutte le definizioni di risorse che presentano un conflitto non vengono importate. • include. Tutte le definizioni di risorsa dai file di importazione vengono importate. È possibile selezionare uno o più tipi di definizione risorsa da escludere dall'importazione selezionando la casella di spunta corrispondente. • exclude. Nessuna definizione di risorsa dai file di importazione viene importata. Gli oggetti importati possono essere modificati perché facciano riferimento a definizioni delle risorse disponibili. <p>Se non viene specificato questo parametro, viene utilizzato il valore predefinito recommended.</p>	

Tabella 51. Parametri di input di `importResource` (Continua)

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>exclude</code>	Facoltativo	Stringa	<p>Definisce i tipi di risorsa esclusi durante l'importazione. È possibile combinare valori multipli in qualsiasi ordine sotto forma di elenco delimitato da punti e virgola. Specificare uno o più tra i seguenti valori:</p> <ul style="list-style-type: none"> • credential esclude le credenziali utente • customproperty esclude le proprietà personalizzate per gli oggetti risorsa. • datasource esclude le definizioni di origine dati • messagedomain esclude i domini del messaggio • notification esclude le definizioni di notifica • servercluster esclude le definizioni di cluster server • server esclude le definizioni di server • topic esclude le definizioni di argomento <p>Se non viene specificato questo parametro, nell'importazione vengono inclusi tutti i tipi.</p>	

Tabella 52. Valore restituito di `importResource`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 53. Eccezioni di `importResource`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine o la cartella di destinazione non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio importa i contenuti del file di esportazione `drafts.pes` nella cartella `/Demo/Drafts` del IBM SPSS Collaboration and Deployment Services Repository

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.importResource(source="C:\\Demo\\drafts.pes", target="/Demo/Drafts")
```

Metodo moveResource

Sposta un file o una cartella in un'altra cartella nel repository. Un file di origine specificato può essere rinominato al momento del suo spostamento, con il tipo di destinazione e l'esistenza che ne determinano il nome finale.

La seguente tabella descrive il comportamento della funzione di ridenominazione al momento dello spostamento di un file:

Tipo di destinazione	La cartella di destinazione esiste	La cartella di destinazione non esiste
cartella	Il file di origine diventa un figlio della cartella di destinazione.	Il file di origine viene spostato nella cartella padre della destinazione specificata e viene rinominato con il nome della cartella di destinazione.
file	Il file di origine viene spostato nella cartella che contiene il file di destinazione e viene rinominato con il nome di destinazione.	Viene riportato un errore.

Ad esempio, se l'origine è il file `/Temp Folder/Temp.txt` e la destinazione specificata è la cartella `/Demo Folder`, è possibile che il risultato sia il seguente:

- La cartella `Demo Folder` esiste, `Temp.txt` viene spostato in `Demo Folder`.
- La cartella `Demo Folder` non esiste, `Temp.txt` viene spostato in `/` e rinominato in `Demo Folder`.

In alternativa, se l'origine `/Temp Folder/Temp.txt` e la destinazione specificata è il file `/Demo Folder/Abc.dat`, è possibile che il risultato sia il seguente:

- La cartella `Demo Folder` esiste, `Temp.txt` viene spostato in `Demo Folder` e rinominato in `Abc.dat`.
- La cartella `Demo Folder` non esiste, viene visualizzato un errore.

```
moveResource(source, target)
```

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	<code>/Temp Folder/Temp.txt</code> o <code>0a58c3670016a7860000010dcee0eaa28219</code>
<code>target</code>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto della cartella in cui spostare il file. È possibile fornire un nuovo nome file per ridenominare la cartella o il file specificato al momento dello spostamento.	<code>/New Folder</code> o <code>/New Folder/abc.dat</code>

Tabella 56. Valore restituito di `moveResource`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 57. Eccezioni di `moveResource`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	L'origine specificata non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio sposta il file `MyReport.rptdesign` dalla cartella `/Demo/Drafts` nella cartella `/Approved`.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.moveResource(source="/Demo/Drafts/MyReport.rptdesign", target="/Approved")
print bSuccess
```

Metodo `removeLabel`

Rimuove un'etichetta dal file nel repository.

```
removeLabel(source, label)
```

Tabella 58. Parametri di input di `removeLabel`

Campo	Utilizzo	Tipo	Valore di esempio	Descrizione
<code>source</code>	Obbligatorio	Stringa	<code>"/Temp Folder/Temp.txt"</code> o <code>"0a58c3670016a7860000010dcee0eaa28219"</code>	Il percorso completo o l'URI dell'oggetto del file nel repository
<code>label</code>	Obbligatorio	Stringa	<code>"Version 1"</code>	Il nome dell'etichetta da rimuovere

Tabella 59. Valore restituito di `removeLabel`

Tipo	Descrizione
String	URI del file aggiornato

Tabella 60. Eccezioni di `removeLabel`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Il seguente esempio rimuove l'etichetta `Draft` dal file `MyReport.rptdesign`.

```
from pes.api.PESImpl import PESImpl
```

```
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
uri = pesImpl.removeLabel(source="/Demo/Drafts/MyReport.rptdesign", label="Draft")
```

Metodo removeSecurity

```
removeSecurity(source,principal,provider,cascade)
```

Rimuove l'ACL (access control list) di sicurezza da un file o da una cartella nel repository.

Tabella 61. Parametri di input di removeSecurity				
Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI dell'oggetto del file o della cartella nel repository	"/Temp Folder/ Temp.txt" o "0a58c3670016a7860000 010dcee0eaa28219"
<i>principal</i>	Obbligatorio	Stringa	L'utente (come ad esempio <i>admin</i>) da rimuovere dalla cartella o dal file specificato.	admin
<i>provider</i>	Facoltativo	Stringa	Il provider di sicurezza (come ad esempio <i>Native</i>) da utilizzare per ottenere le informazioni sull'utente. I seguenti valori sono validi: <ul style="list-style-type: none"> • Native per il provider di sicurezza locale nativo interno al sistema. Questo provider è quello predefinito. • AD_<name> per Active Directory, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema • ADL_<name> per Active Directory con sovrascrittura locale, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema • ldap_<name> per OpenLDAP, dove <name> corrisponde al nome del provider di sicurezza all'interno del sistema 	Native
<i>cascade</i>	Facoltativo	Booleano	Propaga le impostazioni di sicurezza a tutti i file e le cartelle presenti nella cartella specificata.	True o False

Tabella 62. Valore restituito di `removeSecurity`

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Tabella 63. Eccezioni di `removeSecurity`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine o la cartella di destinazione non esiste.
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.
<code>IllegalParameterException</code>	L'utente o il nome del provider di sicurezza specificato non è corretto.

Esempio

Il seguente esempio rimuove l'ACL di un'entità principale dal file `MyReport.rptdesign`.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
bSuccess = pesImpl.removeSecurity(source="/Projects/MyReport.rptdesign", principal="icrod")
```

Metodo `search`

Ricerca i file nel repository, restituendo un elenco di versioni di file con il contenuto dei metadati che corrisponde al criterio di ricerca.

```
search(criteria)
```

Tabella 64. Parametri di input di `search`

Campo	Utilizzo	Tipo	Descrizione	Valore di esempio
<code>criteria</code>	Obbligatorio	Stringa	Il valore utilizzato per cercare i metadati del file	"Age"

Tabella 65. Valore restituito di `search`

Tipo	Descrizione
<code>PageResult</code>	Una struttura in cui ciascuna riga corrisponde ad una corrispondenza nella ricerca. Vedere l'argomento "La classe PageResult" a pagina 58 per ulteriori informazioni.

Tabella 66. Eccezioni di `search`

Tipo	Descrizione
<code>InsufficientParameterException</code>	I parametri obbligatori non sono necessari.

Esempio

Le seguenti ricerche di versioni di file con il testo `Quarterly` in qualsiasi campo dei metadati.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
sResults = pesImpl.search(criteria="Quarterly")
sRows = sResults.getRows()
for sRow in sRows:
    print "Author: ", sRow.getAuthor()
    print "Title: ", sRow.getTitle()
    for child in sRow.getChildRow():
        print "Version: ", child.getVersionMarker()
```

```
print "Label: ", child.getVersionLabel()
print "Keywords:", child.getKeyword()
print "URI:", child.getUri()
```

Il metodo setLabel

Applica un'etichetta ad una versione di un file nel repository. Se il file già dispone di un'etichetta, l'etichetta originale viene sostituita dalla nuova etichetta.

```
setLabel(source, version, label)
```

Tabella 67. Parametri di input per setLabel

Campo	Uso	Tipo	Descrizione	Valore di esempio
source	Obbligatorio	Stringa	Il percorso completo o l'URI oggetto del file nel repository	"/Temp Folder/Temp.txt" oppure "0a58c3670016a7860000010dcee0eaa28219"
version	Obbligatorio	Stringa	La versione specifica del file	"0:2006-08-25 21:15:49.453"
label	Obbligatorio	Stringa	L'etichetta da applicare al file	"Version 1"

Tabella 68. Valore di restituzione per setLabel

Tipo	Descrizione
String	URI del file aggiornato

Tabella 69. Eccezioni per setLabel

Tipo	Descrizione
ResourceNotFoundException	Il file di origine o la versione non esiste.
InsufficientParameterException	Parametri obbligatori non specificati.

Esempio

L'esempio riportato di seguito assegna l'etichetta *Beta* alla seconda versione del file *MyReport.rptdesign*. Il metodo `getVersionMarker` per un oggetto `Resource` restituisce il contrassegno per la versione a cui applicare l'etichetta.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
betaVersion = \
    pesImpl.getAllVersions(source="/Demo/Drafts/MyReport.rptdesign")[1].getVersionMarker()
print "Marker for the beta version is:", betaVersion
uri = pesImpl.setLabel(source="/Demo/Drafts/MyReport.rptdesign", version=betaVersion,
    label="Beta")
```

Il metodo setMetadata

Applica le proprietà dei metadati a file e cartelle nel repository.

La tabella riportata di seguito identifica le proprietà dei metadati ed indica se tali proprietà possono essere applicate a file e cartelle.

Tabella 70. Proprietà dell'oggetto repository

Proprietà dei metadati	Tipo di risorsa
Autore	File

Tabella 70. Proprietà dell'oggetto repository (Continua)

Proprietà dei metadati	Tipo di risorsa
Descrizione	File o cartella
Titolo	File o cartella
Data di scadenza	File o cartella
Parola chiave	File
Argomenti	File
Metadati personalizzati	File o cartella

```
setMetadata(source, version, label, props)
```

Tabella 71. Parametri di input per setMetadata

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo o l'URI oggetto del file o della cartella nel repository	"/Temp Folder/Temp.txt" oppure "0a58c3670016a7860000010dcee0eaa28219"
<i>version</i>	Facoltativo. È possibile specificare la versione o l'etichetta.	Stringa	La versione specifica del file da scaricare	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facoltativo. È possibile specificare la versione o l'etichetta.	Stringa	L'etichetta della versione specifica	"Label 1"

Tabella 71. Parametri di input per setMetadata (Continua)

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>props</i>	Obbligatorio	Dizionario	Contiene tutti i metadati da impostare, nel Dizionario con il nome dei metadati come chiavi. Come illustrato nella colonna Valore di esempio, utilizza l'elenco come valore da topic e Dizionario per customProperty. Per la parte rimanente dei metadati, utilizza stringa.	<pre>{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }</pre>

Tabella 72. Valore di ritorno per setMetadata

Tipo	Descrizione
String	URI del file o della cartella per cui sono stati impostati i metadati

Tabella 73. Eccezioni per setMetadata

Tipo	Descrizione
InsufficientParameterException	Parametri obbligatori non specificati.
ResourceNotFoundException	Il file o la cartella di origine non esiste.

Esempio

L'esempio riportato di seguito assegna la parola chiave *Quarterly* alla versione *Production* del file *MyReport.rptdesign*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
pDict = {'keyword': 'Quarterly'}
uri = pesImpl.setMetadata(source="/Demo/Drafts/MyReport.rptdesign", version=prodVersion,
    props=pDict)
print uri
```

Il metodo uploadFile

Salva un file nel repository dal file system locale, con l'opzione di creare una nuova versione del file se già esiste.

```
uploadFile(source, target, versionFlag)
```

Tabella 74. Parametri di input per uploadFile

Campo	Uso	Tipo	Descrizione	Valore di esempio
<i>source</i>	Obbligatorio	Stringa	Il percorso completo (sul file system locale) del file da caricare	"C:\Temp\Temp.txt"

Tabella 74. Parametri di input per `uploadFile` (Continua)

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>target</code>	Obbligatorio	Stringa	Il percorso completo della cartella di destinazione nel repository in cui deve essere caricato il file	<code>"/Temp Folder"</code>
<code>versionFlag</code>	Facoltativo	Booleano	Se il file specificato già esiste, viene creata una nuova versione del file	<code>True</code> o <code>False</code>

Tabella 75. Valore di ritorno per `uploadFile`

Tipo	Descrizione
String	URI del file caricato

Tabella 76. Eccezioni per `uploadFile`

Tipo	Descrizione
<code>ResourceNotFoundException</code>	Il file di origine o la cartella di destinazione non esiste.
<code>ResourceAlreadyExistsException</code>	Un file o una cartella con lo stesso nome del file di origine già esiste nella cartella di destinazione ed il parametro <code>versionFlag</code> non è specificato.
<code>InsufficientParameterException</code>	Parametri obbligatori non specificati.

Esempio

Questo esempio carica il file `MyReport.rptdesign` nella cartella `/Demo/Drafts` nel repository. Se il file già esiste, viene caricata una nuova versione del file utilizzando il parametro `versionFlag`.

```
from pes.util.PESEExceptions import *
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
try:
    uri = pesImpl.uploadFile(source="C:\Demo\MyReport.rptdesign", target="/Demo/Drafts")
    print "URI for the uploaded file is: ", uri
except ResourceAlreadyExistsException:
    uri = pesImpl.uploadFile(source="C:\Demo\MyReport.rptdesign", target="/Demo/Drafts",
        versionFlag=True)
    print "URI for the uploaded file is: ", uri
```

Classi Wrapper

L'API `PESImpl` include le classi che fungono da wrapper degli oggetti restituiti dai servizi web chiamati dai metodi del web dei contenuti. Tali classi wrapper forniscono un'interfaccia per la visualizzazione delle informazioni restituite dai metodi.

La classe `Resource`

La classe `Resource` agisce come wrapper semplificato per l'oggetto del repository `ResourceSpecifier.Resource`.

Oltre ai metadati standard associati agli oggetti del repository, questa classe comprende tutte le informazioni sui metadati personalizzati definite per gli oggetti nel repository. La [Tabella 77 a pagina 57](#) elenca tutti i metodi disponibili nella classe `Resource`.

Tabella 77. Metodi per la classe Resource

Nome metodo	Descrizione
getAccessControlList	Restituisce un dizionario delle autorizzazioni di sicurezza di un oggetto. Contiene il nome utente come chiave e solo l'autorizzazione più elevata fornita all'utente. Ad esempio: Se l'utente <i>Joe</i> dispone di autorizzazione <i>delete</i> su <i>resource X</i> , <code>getAccessControlList</code> dell'oggetto risorsa che rappresenta <i>X</i> restituisce <code>{ ' Joe ' : ' DELETE ' }</code> e non tutte e tre le autorizzazioni (<i>read</i> , <i>write</i> , <i>delete</i>) dalla chiamata al servizio Web.
getOwner	Restituisce il nome del proprietario dell'oggetto come stringa
getAuthor	Restituisce il nome dell'autore dell'oggetto come stringa
getContentSize	Restituisce le dimensioni dell'oggetto
getCreatedBy	Restituisce il nome dell'utente che ha creato l'oggetto come stringa
getCreationDate	Restituisce la data di creazione dell'oggetto come oggetto <code>datetime</code>
getDescription	Restituisce la descrizione dell'oggetto come elenco
getDescriptionLanguage	Restituisce la lingua dell'oggetto come elenco
getExpirationDate	Restituisce la data di scadenza dell'oggetto come oggetto <code>datetime</code>
isExpired	Indica se l'oggetto specificato è scaduto
getMIMEType	Restituisce il tipo MIME dell'oggetto come stringa
getModificationDate	Restituisce la data dell'ultima modifica dell'oggetto come oggetto <code>datetime</code>
getObjectCreationDate	Restituisce la data di creazione dell'oggetto come oggetto <code>datetime</code>
getObjectLastModifiedBy	Restituisce l'utente che ha modificato l'oggetto per l'ultima volta come stringa
getObjectLastModifiedDate	Restituisce la data dell'ultima modifica all'oggetto come oggetto <code>datetime</code>
getResourceID	Restituisce l'identificativo risorsa dell'oggetto come stringa
getResourcePath	Restituisce il percorso dell'oggetto specificato come stringa
getTitle	Restituisce il titolo per l'oggetto come stringa
getTopicList	Restituisce l'elenco degli argomenti per l'oggetto
getVersionMarker	Restituisce la versione dell'oggetto come stringa
getVersionLabel	Restituisce l'etichetta dell'oggetto come stringa
getCustomMetadata	Restituisce le proprietà personalizzate associate all'oggetto come dizionario
getKeywordList	Restituisce un elenco di parole chiave associate all'oggetto

La classe **IdentificationSpecifier**

Questa classe agisce come wrapper semplificato per l'oggetto del repository `IdentificationSpecifier`, consentendo l'accesso a dati specifici dell'identificazione per l'oggetto.

La [Tabella 78 a pagina 58](#) elenca tutti i metodi disponibili nella classe `IdentificationSpecifier`.

Nome metodo	Descrizione
<code>getIdentifier</code>	Restituisce il valore identificativo di un oggetto come stringa
<code>getVersionMarker</code>	Restituisce la versione di un oggetto come stringa
<code>getVersionLabel</code>	Restituisce l'etichetta applicata ad una versione oggetto come stringa

La classe **PageResult**

Questa classe `PageResult` serve da contenitore per i risultati della ricerca. Una singola voce nei risultati corrisponde ad una riga nell'oggetto `PageResult`.

Ad esempio, una ricerca che restituisce quattro risorse crea un oggetto `PageResult` che contiene quattro righe. La [Tabella 79 a pagina 58](#) elenca tutti i metodi disponibili nella classe `PageResult`.

Nome metodo	Descrizione
<code>getRows</code>	Restituisce un elenco di oggetti <code>SearchRow</code> . Consultare l'argomento " La classe <code>SearchRow</code> " a pagina 58 per ulteriori informazioni.

La classe **SearchRow**

La classe `SearchRow` funziona come contenitore per le informazioni a livello di oggetto relative ad un singolo risultato della ricerca. È possibile accedere ai metadati relativi ad un oggetto utilizzando i metodi di questa classe.

La [Tabella 80 a pagina 58](#) elenca tutti i metodi disponibili nella classe `SearchRow`.

Nome metodo	Descrizione
<code>getTitle</code>	Restituisce il nome del file o della cartella
<code>getAuthor</code>	Restituisce l'autore del file o della cartella
<code>getMIMETYPE</code>	Restituisce il tipo MIME del file
<code>getObjectLastModifiedBy</code>	Restituisce l'utente che ha modificato il file o la cartella per l'ultima volta
<code>getModified</code>	Restituisce la data e l'ora dell'ultima modifica apportata al file o alla cartella
<code>getFolderPath</code>	Restituisce il percorso del file o della cartella
<code>getFolder</code>	Restituisce il nome della cartella principale del file o della cartella
<code>getParentURI</code>	Restituisce l'URI oggetto dell'elemento principale
<code>getTopic</code>	Restituisce gli argomenti associati al file o alla cartella
<code>getChildRow</code>	Restituisce l'elenco degli oggetti <code>SearchChildRow</code> (consultare la seguente sezione per ulteriori informazioni)

Per accedere alle informazioni a livello di versione per un oggetto, utilizzare il metodo `getChildRow` per restituire le righe secondarie corrispondenti alle versioni dell'oggetto.

La classe `SearchChildRow`

La classe `SearchChildRow` funziona come contenitore per le informazioni a livello di versione relative ad un singolo risultato della ricerca. È possibile accedere ai metadati relativi alla versione di un oggetto utilizzando i metodi di questa classe.

La [Tabella 81](#) a pagina 59 elenca tutti i metodi disponibili nella classe `SearchChildRow`.

Tabella 81. Metodi per la classe <code>SearchChildRow</code>	
Nome metodo	Descrizione
<code>getExpirationDate</code>	Restituisce la data di scadenza del file o della cartella
<code>getKeyword</code>	Restituisce le parole chiave associate alla versione del file o della cartella
<code>getVersionLabel</code>	Restituisce l'etichetta della versione del file o della cartella
<code>getDescription</code>	Restituisce la descrizione del file o della cartella
<code>getLanguage</code>	Restituisce la lingua
<code>getVersionCreationDate</code>	Restituisce la data e l'ora di creazione del file o della cartella
<code>getVersionMarker</code>	Restituisce il contrassegno della versione del file o della cartella
<code>getUri</code>	Restituisce l'URI oggetto del file o della cartella

API di gestione dei processi

Lo script di gestione dei processi consente di utilizzare i lavori. Quest'area comprende le seguenti funzionalità:

- Esecuzione dei lavori
- Richiamo della cronologia del lavoro
- Richiamo dei dettagli del lavoro

Questa sezione illustra i metodi `PESImpl` utilizzati per la gestione dei lavori memorizzati nel repository. Ciascun metodo contiene informazioni dettagliate sulla sintassi, un esempio ed i messaggi previsti.

Metodi

Le sezioni riportate di seguito elencano tutti i metodi di creazione script di gestione dei processi supportati per IBM SPSS Collaboration and Deployment Services.

Nota: per tutti i metodi che richiedono un percorso a file/cartelle nel repository, è possibile utilizzare il percorso o l'URI oggetto. È possibile ottenere l'URI oggetto visualizzando le proprietà dell'oggetto in IBM SPSS Deployment Manager.

Il metodo `cancelJob`

Annulla un lavoro in esecuzione.

```
cancelJob(executionId)
```

Tabella 82. Parametri di input per cancelJob

Campo	Uso	Tipo	Descrizione	Valore di esempio
executionId	Obbligatorio	Stringa	ID di esecuzione per il lavoro	0a58c33d002ce9080000010e0ccf7b01800e

Tabella 83. Valore di ritorno per cancelJob

Tipo	Descrizione
Boolean	Restituisce un messaggio che indica l'esito positivo quando il messaggio viene annullato

Esempio

Questo esempio termina l'esecuzione del lavoro *Reports*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
execId = pesImpl.executeJob(source='/Demo/Jobs/Reports', notification = True,
    asynchronous=True)
print "Execution ID: ", execId
status = pesImpl.cancelJob(execId)
print "Successful cancellation: ", status
```

Il metodo deleteJobExecutions

Elimina una o più esecuzioni lavoro.

```
deleteJobExecutions(executionId)
```

Tabella 84. Parametri di input per deleteJobExecutions

Campo	Uso	Tipo	Descrizione	Valore di esempio
executionId	Obbligatorio	Elenco	Elenco di identificativi di esecuzione lavoro	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

Tabella 85. Valore di ritorno per deleteJobExecutions

Tipo	Descrizione
Boolean	True o False, a seconda che il metodo venga eseguito correttamente o meno.

Esempio

Questo esempio elimina le esecuzioni per il lavoro *Reports*.

```
from pes.util.PESExceptions import *
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
executions = pesImpl.getJobExecutionList(source="/Demo/Jobs/Reports")
execRows = executions.getRows()

# Get the execution ID from the execution history
deletelist = []
for exrow in execRows :
    uuid = exrow.getEventObjId()
    deletelist.append(uuid)

if len(deletelist) != 0:
    print 'Deleting ',len(deletelist) , ' histories'
    pesImpl.deleteJobExecutions(deletelist)
```

Il metodo `executeJob`

Esegue un lavoro in modo sincrono o asincrono in base ai parametri trasmessi. In caso di esecuzione sincrona, il metodo non restituisce il controllo fino a quando il lavoro non viene completato. In caso di esecuzione asincrona, il metodo restituisce il controllo dopo l'avvio del lavoro.

```
executeJob(source,notification,asynchronous)
```

Tabella 86. Parametri di input per `executeJob`

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	Stringa	Il percorso completo (sul file system locale) del file da caricare	"C:\Temp\Temp.txt"
<code>notification</code>	Facoltativo	Booleano	Indica se il lavoro viene eseguito con o senza notifiche. Il valore predefinito è False.	True o False
<code>asynchronous</code>	Facoltativo	Booleano	Indica se il lavoro viene eseguito in modo asincrono. Il valore predefinito è False.	True o False

Tabella 87. Valore di ritorno per `executeJob`

Tipo	Descrizione
String	L'identificativo univoco per l'esecuzione del lavoro. Tale identificativo viene utilizzato per fare riferimento ad una particolare esecuzione lavoro.

Esempio

Questo esempio avvia l'esecuzione del lavoro `Reports` in modo asincrono con le notifiche.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
execId = pesImpl.executeJob(source='/Demo/Jobs/Reports', notification = True,
                           asynchronous=True)
print "Execution ID: ", execId
```

Il metodo `getJobExecutionDetails`

Elenca i dettagli di esecuzione per un lavoro specifico, comprese le interazioni e le fasi del lavoro.

```
getJobExecutionDetails(executionId,log,target)
```

Tabella 88. Parametri di input per `getJobExecutionDetails`

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>executionId</code>	Obbligatorio	Stringa	L'ID di esecuzione del lavoro	0a58c33d002ce9080000010e0ccf7b01800e
<code>log</code>	Facoltativo	Booleano	Indica se il log del lavoro è visualizzato in linea	True o False
<code>target</code>	Facoltativo	Stringa	Il percorso (sul file system locale) per memorizzare i log. Utilizzato solo insieme al parametro <code>--log</code> .	"c:\logs"

Tabella 89. Valore di ritorno per `getJobExecutionDetails`

Tipo	Descrizione
<code>jobExecutionDetails</code>	Dettagli relativi ad un'esecuzione lavoro. Consultare l'argomento “La classe <code>jobExecutionDetails</code>” a pagina 63 per ulteriori informazioni.

Esempio

Questo esempio richiama le informazioni relative alle esecuzioni delle fasi del lavoro per l'esecuzione lavoro con identificativo `execId`, inviando alla console il risultato per ciascuna fase.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
execDetails = pesImpl.getJobExecutionDetails(executionId=execId)
print "Job ID: ", execDetails.getUUID()
print "Event ID: ", execDetails.getEventUUID()
print "Started: ", execDetails.getStartDateTime()
print "Ended: ", execDetails.getEndDateTime()
for step in execDetails.getJobStepDetails():
    print "Step ID: ", step.getEventUUID()
    print "Step Name: ", step.getEventName()
    print "Started: ", step.getStartDateTime()
    print "Ended: ", step.getEndDateTime()
    print "Success: ", step.getExecutionSuccess()
```

Il metodo `getJobExecutionList`

Elenca le esecuzioni per un lavoro specifico, compresi i lavori completati e quelli attualmente in esecuzione, per tutti le versioni del lavoro.

```
getJobExecutionList(source)
```

Tabella 90. Parametri di input per `getJobExecutionList`

Campo	Uso	Tipo	Descrizione	Valore di esempio
<code>source</code>	Obbligatorio	Stringa	Il percorso completo del lavoro nel repository.	<code>"/testJob"</code>

Tabella 91. Valore di ritorno per `getJobExecutionList`

Tipo	Descrizione
<code>PageResult</code>	Contenitore per l'elenco di esecuzioni per un lavoro. Consultare l'argomento “La classe <code>PageResult</code>” a pagina 63 per ulteriori informazioni. .

Esempio

Questo esempio richiama le esecuzioni per il lavoro `Reports`, inviando alla console le informazioni relative a ciascuna esecuzione.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
executions = pesImpl.getJobExecutionList(source="/Demo/Jobs/Reports")
execRows = executions.getRows()
if execRows:
    for exrow in execRows:
        print "Job Path: ", exrow.getPath()
        print "Object ID: ", exrow.getObjId()
        print "Event ID: ", exrow.getEventObjId()
        print "Version ", exrow.getVersionMarker()
        print "Started: ", exrow.getEventStartDateTime()
        print "Ended: ", exrow.getEventEndDateTime()
```

Classi wrapper

L'API `PESImpl` comprende le classi che vengono utilizzate come wrapper per gli oggetti restituiti dai servizi Web richiamati dai metodi di gestione dei processi. Tali classi wrapper forniscono un'interfaccia per la visualizzazione delle informazioni restituite dai metodi.

La classe PageResult

Questa classe PageResult serve da contenitore per i risultati dell'esecuzione del lavoro, consentendo il richiamo di dati specifici dell'esecuzione del lavoro.

Una singola esecuzione del lavoro corrisponde ad una riga nell'oggetto PageResult. Ad esempio, un lavoro eseguito quattro volte corrisponde a un oggetto PageResult che contiene quattro righe. La Tabella 92 a pagina 63 elenca tutti i metodi disponibili nella classe PageResult.

Tabella 92. Metodi per la classe PageResult	
Nome metodo	Descrizione
getRows	Restituisce un elenco di oggetti Row, ciascuno dei quali rappresenta un'esecuzione di un lavoro. Consultare l'argomento “La classe Row” a pagina 63 per ulteriori informazioni.

La classe Row

La classe Row serve da contenitore per le informazioni a livello del lavoro relative all'esecuzione di un lavoro. È possibile accedere ai metadati relativi all'esecuzione di un lavoro utilizzando i metodi di questa classe.

La Tabella 93 a pagina 63 elenca tutti i metodi disponibili nella classe Row.

Tabella 93. Metodi per la classe Row	
Nome metodo	Descrizione
getObjId	Restituisce l'ID di esecuzione del lavoro
getPath	Restituisce il percorso del lavoro
getVersionMarker	Restituisce il contrassegno della versione del lavoro eseguito
getVersionLabel	Restituisce l'etichetta della versione del lavoro eseguito
getEventObjId	Restituisce l'ID evento del lavoro eseguito
getEventState	Restituisce lo stato del lavoro in esecuzione
getEventCompletionCode	Restituisce il codice di completamento del lavoro
getEventStartDateTime	Restituisce la data e l'ora di inizio del lavoro
getEventEndDateTime	Restituisce la data e l'ora di fine del lavoro
getQueuedDateTime	Restituisce la data e l'ora di accodamento del lavoro

La classe jobExecutionDetails

Questa classe è restituita dal metodo getJobExecutionDetails. Memorizza i dettagli di esecuzione per un lavoro ed include un elenco di oggetti jobStepExecution che forniscono informazioni relative a ciascuna fase nel lavoro.

La Tabella 94 a pagina 63 elenca tutti i metodi disponibili nella classe jobExecutionDetails.

Tabella 94. Metodi per la classe jobExecutionDetails	
Nome metodo	Descrizione
getJobStepDetails	Restituisce un elenco di oggetti jobStepExecutionDetails. Consultare l'argomento “La classe jobStepExecutionDetails” a pagina 64 per ulteriori informazioni.
getArtifactLocation	Restituisce un elenco di percorsi delle risorse lavoro

Tabella 94. Metodi per la classe `jobExecutionDetails` (Continua)

Nome metodo	Descrizione
<code>getCompletionCode</code>	Restituisce il codice di completamento dell'esecuzione lavoro
<code>getEndDateTime</code>	Restituisce la data e l'ora di fine dell'esecuzione lavoro
<code>getEventName</code>	Restituisce il nome evento dell'esecuzione lavoro
<code>getEventUUID</code>	Restituisce l'ID evento dell'esecuzione lavoro
<code>getExecutionState</code>	Restituisce lo stato di esecuzione dell'esecuzione lavoro
<code>getExecutionSuccess</code>	Restituisce lo stato di errore o di esito positivo dell'esecuzione lavoro
<code>getExecutionWarning</code>	Indica se sono presenti avvisi
<code>getLog</code>	Restituisce il log generato (come stringa)
<code>getNotificationEnabled</code>	Indica se sono abilitate le notifiche email
<code>getQueuedDateTime</code>	Restituisce la data e l'ora di accodamento dell'esecuzione lavoro
<code>getStartDateTime</code>	Restituisce la data e l'ora di inizio dell'esecuzione lavoro
<code>getUserName</code>	Restituisce il nome dell'utente che ha eseguito il lavoro
<code>getUUID</code>	Restituisce l'ID di esecuzione del lavoro

La classe `jobStepExecutionDetails`

Questa classe memorizza i dettagli di esecuzione per una fase del lavoro e memorizza un elenco degli oggetti `jobStepChildExecutionDetails`. Questa classe contiene l'oggetto `ExecutionDetails`, a cui delega tutte le chiamate ai metodi.

La Tabella 95 a pagina 64 elenca tutti i metodi disponibili nella classe `jobStepExecutionDetails`.

Tabella 95. Metodi per la classe `jobStepExecutionDetails`

Nome metodo	Descrizione
<code>getJobStepChldExecutionList</code>	Restituisce un elenco di oggetti <code>jobStepChildExecutionDetails</code> . Consultare l'argomento "La classe <code>jobStepChildExecutionDetails</code> " a pagina 65 per ulteriori informazioni.
<code>getArtifactLocation</code>	Restituisce un elenco di percorsi delle risorse della fase lavoro
<code>getCompletionCode</code>	Restituisce il codice di completamento della fase lavoro
<code>getEndDateTime</code>	Restituisce la data e l'ora di fine della fase lavoro
<code>getEventName</code>	Restituisce il nome evento della fase lavoro
<code>getEventUUID</code>	Restituisce l'ID evento della fase lavoro
<code>getExecutionState</code>	Restituisce lo stato di esecuzione della fase lavoro

Tabella 95. Metodi per la classe `jobStepExecutionDetails` (Continua)

Nome metodo	Descrizione
<code>getExecutionSuccess</code>	Restituisce lo stato di errore o di esito positivo della fase lavoro
<code>getExecutionWarning</code>	Indica se sono presenti avvisi
<code>getLog</code>	Restituisce il log generato (come stringa)
<code>getNotificationEnabled</code>	Indica se sono abilitate le notifiche email
<code>getQueuedDateTime</code>	Restituisce la data e l'ora di accodamento della fase lavoro
<code>getStartDateTime</code>	Restituisce la data e l'ora di inizio della fase lavoro
<code>getUserName</code>	Restituisce il nome dell'utente che ha eseguito la fase lavoro
<code>getUUID</code>	Restituisce l'ID di esecuzione della fase lavoro

La classe `jobStepChildExecutionDetails`

La classe `jobStepChildExecutionDetails` funziona come contenitore per le esecuzioni secondarie delle singole fasi del lavoro. Ad esempio, una fase di lavoro del report iterativa produce un'esecuzione secondaria per ciascuna iterazione della fase. È possibile accedere ai metadati relativi alle esecuzioni secondarie utilizzando i metodi di questa classe.

La Tabella 96 a pagina 65 elenca tutti i metodi disponibili nella classe `jobStepChildExecutionDetails`.

Tabella 96. Metodi per la classe `jobStepChildExecutionDetails`

Nome metodo	Descrizione
<code>getArtifactLocation</code>	Restituisce un elenco dei percorsi delle risorse di esecuzione secondarie
<code>getCompletionCode</code>	Restituisce il codice di completamento dell'esecuzione secondaria
<code>getEndDateTime</code>	Restituisce la data e l'ora di fine dell'esecuzione secondaria
<code>getEventName</code>	Restituisce il nome evento dell'esecuzione secondaria
<code>getEventUUID</code>	Restituisce l'ID evento dell'esecuzione secondaria
<code>getExecutionState</code>	Restituisce lo stato di esecuzione dell'esecuzione secondaria
<code>getExecutionSuccess</code>	Restituisce lo stato di errore o di esito positivo dell'esecuzione secondaria
<code>getExecutionWarning</code>	Indica se sono presenti avvisi
<code>getLog</code>	Restituisce il log generato (come stringa)
<code>getNotificationEnabled</code>	Indica se sono abilitate le notifiche email
<code>getQueuedDateTime</code>	Restituisce la data ed ora di accodamento dell'esecuzione secondaria
<code>getStartDateTime</code>	Restituisce la data ed ora di inizio dell'esecuzione secondaria

Tabella 96. Metodi per la classe `jobStepChildExecutionDetails` (Continua)

Nome metodo	Descrizione
<code>getUserName</code>	Restituisce il nome dell'utente che ha eseguito l'esecuzione secondaria
<code>getUUID</code>	Restituisce l'ID di esecuzione dell'esecuzione secondaria

Script di esempio

Gli script di esempio che illustrano l'utilizzo della classe `PESImpl` sono installati nella seguente directory:

`<installation location>/samples`

Tali script eseguono diverse attività, incluse quelle riportate di seguito:

- Eliminazione degli elementi scaduti da IBM SPSS Collaboration and Deployment Services Repository
- Eliminazione delle risorse inoltrate scadute
- Eliminazione delle cronologie lavoro

È possibile richiamare gli script da una fase lavoro Generale in IBM SPSS Deployment Manager per eseguire le attività di manutenzione del repository.

Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti. Questo materiale potrebbe essere disponibile da IBM in altre lingue. Tuttavia, potrebbe essere necessario disporre di una propria copia del prodotto o versione di prodotto in quella lingua per potervi accedere.

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
Deutschland*

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

IBM (INTERNATIONAL BUSINESS MACHINES CORPORATION) FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcune giurisdizioni non escludono le garanzie implicite; di conseguenza la suddetta esclusione potrebbe, in questo caso, non essere 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.

Tutti i riferimenti a siti Web non IBM sono forniti unicamente a scopo di consultazione e non devono essere in alcun modo considerati come complementari a 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 qualsiasi informazione fornita dall'utente nel modo che ritiene più idoneo senza incorrere in alcun obbligo nei confronti dell'utente stesso.

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

*IBM Director of Licensing
IBM Europe
Schoenaicher Str. 220*

D-7030 Boeblingen
Deutschland

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.

I dati delle prestazioni e gli esempi client citati vengono presentati solo a scopo illustrativo. I risultati delle prestazioni effettive possono variare in base alle configurazioni specifiche e alle condizioni di funzionamento.

Le informazioni relative a prodotti non IBM sono ottenute dai fornitori di quei prodotti, dagli annunci pubblicati o da altre fonti disponibili al pubblico. IBM non ha testato quei prodotti e non può garantire l'accuratezza delle prestazioni, la compatibilità o qualsiasi altra dichiarazione relativa a prodotti non IBM. Commenti relativi alle prestazioni di prodotti non IBM, dovrebbero essere indirizzati ai fornitori di questi prodotti.

Qualsiasi affermazione relativa agli obiettivi e alla direzione futura di IBM è soggetta a modifica o revoca senza preavviso e concerne esclusivamente gli scopi dell'azienda.

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 a persone o aziende commerciali reali è puramente casuale.

LICENZA SUL COPYRIGHT:

Le presenti informazioni includono programmi di applicazioni di esempio nel linguaggio sorgente che illustrano le tecniche di programmazione sulle diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento a IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. I programmi di esempio non sono stati completamente testati in ogni condizione. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di tali programmi. I programmi di esempio vengono forniti "COSÌ COME SONO", senza garanzia di alcun tipo. IBM declina ogni responsabilità derivante dall'uso degli stessi.

Considerazioni relative alla politica sulla privacy

I prodotti IBM Software, incluse le soluzioni Software as a Service ("Offerte software") possono utilizzare cookie o altre tecnologie per raccogliere informazioni sull'utilizzo del prodotto, per consentire di migliorare l'esperienza dell'utente finale, per personalizzare le interazioni con l'utente finale o per altri scopi. In molti casi l'Offerta software non raccoglie alcuna informazione personale. Alcune delle Offerte software possono abilitare la raccolta di informazioni personali. Se questa Offerta software utilizza i cookie per raccogliere dati personali, specifiche informazioni sull'utilizzo dei cookie di questa offerta sono riportate di seguito.

Questa Offerta software non utilizza cookie o altre tecnologie per raccogliere informazioni che consentono l'identificazione personale.

Se le configurazioni distribuite per questa offerta software forniscono all'utente, in qualità di cliente, la possibilità di raccogliere personalmente informazioni identificabili dagli utenti finali mediante cookie e altre tecnologie, l'utente dovrebbe richiedere una consulenza legale relativa alle leggi applicabili in materia di raccolta di tali dati incluso i requisiti di preavviso e consenso.

Per ulteriori informazioni sull'uso a tale scopo delle diverse tecnologie, inclusi i cookie, fare riferimento alla politica della privacy IBM all'indirizzo <http://www.ibm.com/privacy> e alla dichiarazione sulla privacy online IBM all'indirizzo <http://www.ibm.com/privacy/details>, la sezione "Cookies, Web Beacons and Other

Technologies" e la sezione "IBM Software Products and Software-as-a-Service Privacy Statement" all'indirizzo <http://www.ibm.com/software/info/product-privacy>.

Marchi

IBM, il logo IBM e [ibm.com](http://www.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. Per un elenco aggiornato di marchi IBM, consultare il web nella sezione Copyright and trademark information, all'indirizzo www.ibm.com/legal/copytrade.shtml.

Adobe, il logo Adobe, PostScript e il logo PostScript sono marchi o marchi registrati di Adobe Systems Incorporated negli Stati Uniti e/o in altri paesi.

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.

Nomi di altri prodotti e servizi possono essere marchi di IBM o di altre società.

