

IBM SPSS Collaboration and Deployment Ser-
vices - Essentials for Python
Version 8 Release 2

Handbuch für Entwickler



Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 69 gelesen werden.

Produktinformation

Diese Ausgabe bezieht sich auf Version 8, Release 2, Modifikation 2 von IBM® SPSS Collaboration and Deployment Services und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

© Copyright International Business Machines Corporation 2000, 2020.

Inhaltsverzeichnis

Kapitel 1. IBM SPSS Collaboration and Deployment Services - Essentials for Python	1
Übersicht.....	1
Installation.....	1
Veraltete Features.....	1
Kapitel 2. Befehlszeilenscripts.....	3
Globale Schlüsselwörter.....	3
Repository-Verbindungen.....	4
Content-Repository-Scripts.....	5
Schlüsselwörter.....	5
Content-Repository-Operationen.....	6
Prozessmanagementfunktionen.....	23
Schlüsselwörter.....	24
Prozessmanagementoperationen.....	24
Kapitel 3. PESImpl-Modul.....	27
Content-Repository-API.....	28
Methoden.....	28
Wrapperklassen.....	58
Prozessmanagement-API.....	61
Methoden.....	61
Wrapperklassen.....	64
Beispielscripts.....	68
Bemerkungen.....	69
Hinweise zur Datenschutzrichtlinie	70
Marken.....	71

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

Übersicht

IBM SPSS Collaboration and Deployment Services stellt ein Scripting-Framework mit einigen APIs bereit, mit denen erfahrene Benutzer und Administratoren eigenständige Routinen oder Stapeljobs schreiben können, die eine Gruppe von Routinen zum Arbeiten mit Repository-Objekten und -Jobs miteinander kombinieren. Dadurch können Massenaufgaben stark vereinfacht werden, zu denen etwa die folgenden zählen:

- Ändern von Sicherheitsberechtigungen für große Benutzergruppen
- Bezeichnen oder Entfernen von Beschriftungen von einer großen Anzahl an Ordnern oder Dateien
- Hoch- oder Herunterladen einer großen Anzahl an Ordnern oder Dateien

Das Framework umfasst die Möglichkeit, Aufgaben über die Befehlszeile auszuführen, sowie eine funktionsreiche API für die Interaktion mit IBM SPSS Collaboration and Deployment Services Repository in Ihrem eigenen Python-Knoten.

Allgemeine Informationen zu Python, einer dynamischen, objektorientierten Programmiersprache, finden Sie auf der [Python-Website](#).

Installation

Das Scripting-Framework kann auf Windows- und UNIX-Plattformen installiert werden. Die Scriptplattform ist unabhängig von der Plattform, die vom Repository verwendet und auf die von der Scriptfunktion zugegriffen wird.

Ein Repository, das z. B. auf der Windows-Plattform ausgeführt wird, kann von Scriptfunktionen aufgerufen werden, die auf der UNIX-Plattform ausgeführt werden.

Installationsanweisungen finden Sie im Installationshandbuch zu IBM SPSS Collaboration and Deployment Services - Essentials for Python.

Veraltete Features

Wenn Sie von einem früheren Release von IBM SPSS Collaboration and Deployment Services migrieren, müssen Sie beachten, dass viele Features seit der letzten Version veraltet sind und nicht mehr verwendet werden.

Wenn ein Feature veraltet ist, entfernt IBM Corp. dieses Feature möglicherweise in einem nachfolgenden Release des Produkts. Zukünftige Investitionen werden sich auf die unter der empfohlenen Migrationsaktion aufgelistete strategische Funktion konzentrieren. In der Regel wird ein Feature nur dann nicht mehr verwendet, wenn es eine funktional entsprechende Alternative gibt.

In diesem Release wurden keine Features nicht weiter unterstützt. Für Referenzzwecke enthält die folgende Tabelle Features, die in neueren Vorgängerversionen des Produkts nicht weiter unterstützt wurden. Sofern möglich, ist in der Tabelle auch die empfohlene Migrationsaktion angegeben.

Tabelle 1. Veraltete Features aus Vorgängerversionen

Einstellung der Unterstützung	Empfohlene Migrationsaktion
Sicherheitsprovider: Active Directory mit lokaler Übersteuerung, wodurch erweiterte Gruppen und berechtigte Benutzer unterstützt werden	Active Directory-Standardsicherheitsprovider mit gegebenenfalls hinzugefügten erforderlichen Gruppen verwenden
IBM SPSS Collaboration and Deployment Services Enterprise View	Analysedatenansicht verwenden
IBM SPSS Collaboration and Deployment Services Enterprise View Driver	Analysedatenansicht verwenden
Szenariodateien	Szenariodateien (.scn) werden nicht weiter unterstützt. Enterprise View-Quellenknoten können in Deployment Manager nicht geändert werden. Alte Szenariodateien können im IBM SPSS Modeler-Client geändert und als Datenstromdateien erneut gespeichert werden. Auch Scoring-Konfigurationen, die eine Szenariodatei verwendet haben, müssen gelöscht und basierend auf einer Datenstromdatei erneut erstellt werden.
Webinstallation für IBM SPSS Deployment Manager	Standalone-Installationsprogramm verwenden
BIRT Report Designer for IBM SPSS	Keine
Viewer von BIRT Report Designer for IBM SPSS	Keine
IBM SPSS Collaboration and Deployment Services Portlet	IBM SPSS Collaboration and Deployment Services Deployment Portal direkt oder die Web-Service-APIs verwenden
IBM SPSS Collaboration and Deployment Services Web Part	IBM SPSS Collaboration and Deployment Services Deployment Portal direkt oder die Web-Service-APIs verwenden
API von Scoring-Service Version 1	API von Scoring-Service Version 2
Zeitplanungsservice	Keine
Berichterstellungsservice	Keine
Operation login des Authentifizierungsservice	Operation doLogin des Authentifizierungsservice
Operation search des Suchservice	Operation search2.5 des Suchservice
JAR-Datei für SPSS AXIS/Castor-Web-Service-Client	Mit Java Runtime Environment, der integrierten Entwicklungsumgebung (IDE) oder Eclipse Web Tools Platform (WTP) bereitgestellte Tools verwenden
API-Funktion clemrtl_setLogFile()	Keine

Kapitel 2. Befehlszeilenscripts

Die Python-Datei `CADSTool.py` kann über die Befehlszeile verwendet werden, um in IBM SPSS Collaboration and Deployment Services Repository gespeicherte Ressourcen zu bearbeiten.

Die allgemeine Syntax, die verwendet wird, um Scriptoperationen von IBM SPSS Collaboration and Deployment Services über die Befehlszeile aufzurufen, ist:

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

Erläuterung:

- `<Operation>` bezeichnet die aufzurufende Funktion
- `<Schlüsselwörter>` definiert Schlüsselwörter/Wertepaare, die als Eingabeparameter für die Funktion verwendet werden

Globale Schlüsselwörter

In Tabelle 2 auf Seite 3 werden die Schlüsselwörter aufgelistet, die von allen Scripterstellungsfunktionen von IBM SPSS Collaboration and Deployment Services unterstützt werden. Die zweite Spalte enthält optionale, abgekürzte Versionen der Schlüsselwörter. Bei den Schlüsselwörtern wird zwischen Groß- und Kleinschreibung unterschieden.

Schlüsselwort	Optionale Kurzversion	Verwendung
<code>--user</code>	<code>-u</code>	<p>Der Benutzername für die Verbindung zum Repository-Server. Der Wert muss ein Präfix enthalten, das den Sicherheitsprovider angibt, wenn es sich nicht um einen Benutzer des nativen Providers handelt. Die folgenden Präfixwerte sind gültig:</p> <ul style="list-style-type: none">• "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider.• <code>AD_<name></code> für Active Directory, wobei <code><name></code> dem Namen des Sicherheitsproviders im System entspricht.• <code>ADL_<name></code> für Active Directory mit lokalem Überschreiben, wobei <code><name></code> dem Namen des Sicherheitsproviders im System entspricht.• <code>ldap_<name></code> für OpenLDAP, wobei <code><name></code> dem Namen des Sicherheitsproviders im System entspricht. <p>Fügen Sie nach dem Präfix einen Schrägstrich und den Benutzernamen ein. Schließen Sie für Active Directory-Provider die Domäne in das Präfix ein. Für den Benutzer <code>icrod</code> in der Domäne <code>MYDOMAIN</code> der Active Directory-Instanz <code>AD_SPSSAD</code> lautet der Wert beispielsweise <code>AD_SPSSAD/MYDOMAIN/icrod</code>. Wenn der Benutzer <code>icrod</code> in der Domäne <code>SPSSLDAP</code> des OpenLDAP-Providers befindet, lautet der Wert <code>ldap_SPSSLDAP/icrod</code>.</p>
<code>--password</code>	<code>-p</code>	<p>Das Kennwort für die Verbindung zum Repository-Server.</p>

Tabelle 2. Globale Schlüsselwörter (Forts.)

Schlüsselwort	Optionale Kurzversion	Verwendung
--host	-q	Der Name des Hosts/Servers, auf dem das Repository installiert ist
--port	-o	Die Portnummer des Repository-Servers
--ssl		Gibt an, dass der Repository-Server das SSL-Protokoll (Secure Socket Layer) zur Verschlüsselung der Kommunikation verwendet. Bei Verwendung dieses Schlüsselworts muss der Repository-Server für SSL konfiguriert sein. Weitere Informationen finden Sie in der Administratordokumentation.
--server_url	-S	Vollständige URL des Repository-Servers. Verwenden Sie dieses Schlüsselwort, wenn die URL des Servers ein benutzerdefiniertes Kontextstammelement enthält. Wenn Sie die Server-URL angeben, sind für die Schlüsselwörter host, port und ssl keine Werte erforderlich.
--useDefault	-z	Verwendet die in der Datei <code>Authorization.properties</code> definierten Serververbindungsinformationen
--help	-h	Hilfeinformationen zum Scripting-Modul

Repository-Verbindungen

Sie müssen die Benutzer-ID, das Kennwort und die Repository-Serverinformationen von IBM SPSS Collaboration and Deployment Services Repository am Ende jedes Befehls angeben.

Folgende Methoden können verwendet werden, um diese Verbindungsinformationen bereitzustellen:

- Die Verwendung von Schlüsselwörtern, wie in den folgenden Beispielen:

```
--user Benutzer --password Kennwort --host Host --port Port
```

```
--user Benutzer --password Kennwort --server_url URL
```

- Über die Datei `Authorization.properties`, wobei der Befehl einen Parameter `--useDefault` (oder die Kurzversion `-z`) enthält. Dabei werden die Verbindungsinformationen aus der Datei `Authorization.properties` abgerufen, die unter `Scriptordner\Lib\site-packages\config\Authorization.properties` zu finden ist. Verwenden Sie einen einfachen Texteditor, um folgende Werte in der Datei zu ändern und an die Einstellungen Ihres Repository anzupassen:

```
# Autorisierungsinformationen
user=admin
password=kwf
host=IhrHost
port=80
```

Alternativ können Sie das Schlüsselwort `server_url` in der Eigenschaftendatei verwenden.

```
# Autorisierungsinformationen
user=admin
password=kwf
server_url=http://IhrHost:80/Kontextstammverzeichnis
```

Parameter, die über die Befehlszeile übergeben werden, haben stets Vorrang. Wenn beispielsweise `--user` und `--password` über die Befehlszeile angegeben werden und zudem der Parameter `--useDefault` oder `-z` angegeben wird, werden der Benutzer und das Kennwort aus der Befehlszeile verwendet und der Host und Port werden aus der Datei `Authorization.properties` abgerufen. Wenn dagegen

der Benutzer, das Kennwort, der Host und Port über die Befehlszeile angegeben werden und zudem der Parameter `--useDefault` oder `-z` verwendet wird, wird der Parameter `--useDefault` ignoriert und es werden nur die Befehlszeileninformationen verwendet.

Bei allen hier beschriebenen APIs wird in der Syntax und in den Beispielen der Parameter `-z` verwendet, um die Anzahl der erforderlichen Parameter auf ein Minimum zu beschränken.

Content-Repository-Scripts

Content-Repository-Scripts bieten die Möglichkeit, Repository-Ressourcen wie Dateien und Ordner zu verwenden. Dieser Bereich enthält folgende Funktionen:

- Erstellen und Löschen von Ordnern
- Hoch- und Herunterladen von Dateien
- Exportieren und Importieren von Ordnern
- Verwalten von Beschriftungen, Sicherheit und Metadaten

In diesem Abschnitt wird die Python-Befehlszeilenverwendung von Scripts für Repository-Funktionen beschrieben. Jede Operation enthält detaillierte Syntaxinformationen, ein Beispiel und erwartete Nachrichten.

Schlüsselwörter

In Tabelle 3 auf Seite 5 sind die für Repository-Funktionen unterstützten Schlüsselwörter aufgelistet. Die zweite Spalte enthält optionale, abgekürzte Versionen der Schlüsselwörter.

Wichtig: Bei Schlüsselwörtern wird zwischen Groß- und Kleinschreibung unterschieden.

Schlüsselwort	Optionale Kurzversion	Verwendung
<code>--source</code>	<code>-s</code>	Der Pfad der Quelldatei oder des Quellenordners
<code>--target</code>	<code>-t</code>	Der Pfad des Zielordners
<code>--version</code>	<code>-v</code>	Die Version einer Datei
<code>--principal</code>	<code>-r</code>	Der Benutzer, dem die Berechtigung zugewiesen werden muss
<code>--permission</code>	<code>-n</code>	Der Berechtigungstyp (z. B. Lesen, Schreiben, Ändern, Löschen)
<code>--label</code>	<code>-l</code>	Die Beschriftung, die einer Version einer Datei zugeordnet wird
<code>--criteria</code>	<code>-c</code>	Die Suchkriterien für die Suche nach Metadatenattributen von Dateien oder Ordnern
<code>--author</code>	<code>-a</code>	Der Name des Autors einer Datei oder eines Ordners
<code>--description</code>	<code>-d</code>	Die Beschreibung für eine Datei oder einen Ordner
<code>--title</code>	<code>-i</code>	Der Titel für eine Datei oder einen Ordner
<code>--expirationDate</code>	<code>-q</code>	Das Ablaufdatum für eine Datei oder einen Ordner
<code>--expirationStartDate</code>		Das Ablaufanfangsdatum für eine Datei oder einen Ordner
<code>--expirationEndDate</code>		Das Ablaufenddatum für eine Datei oder einen Ordner

Tabelle 3. Schlüsselwörter für Repository-APIs (Forts.)

Schlüsselwort	Optionale Kurzversion	Verwendung
--keyword	-k	Das Schlüsselwort für eine Datei oder einen Ordner
--cascade	-x	Gibt an, dass Sicherheitseinstellungen für einen Ordner an Unterordner und Dateien weitergegeben werden sollen
--provider	-f	Der Sicherheitsanbieter, der zum Abruf der Principals verwendet wird
--createVersion	-b	Gibt an, dass eine neue Version einer Datei erstellt werden soll
--contentLanguage	-g	Die Inhaltssprache für eine Datei oder einen Ordner
--topic		Die Themen, die einer Datei oder einem Ordner zugeordnet sind. Sie können mehrere Werte eingeben, z. B. --topic "Thema1;Thema2"
--modifiedBy		Der Benutzer, der eine Datei oder einen Ordner geändert hat
--mimeType		Der MIME-Typ einer Datei
--createdBy		Der Benutzer, der eine Datei oder einen Ordner erstellt hat
--submittedHierarchy		Gibt an, ob im Ordner <i>Submitted Jobs</i> gesucht werden soll.
--propertyName		Der Name einer benutzerdefinierten Eigenschaft
--customProperty		Der Name/Wert-Paar einer benutzerdefinierten Eigenschaft, die aktualisiert werden soll
--propertyName		Der Name der benutzerdefinierten Eigenschaft, für die gültige Werte abgerufen werden sollen

Für alle Operationen, die Beschriftungs- und Versionsinformationen akzeptieren, sollte der Benutzer entweder eine Beschriftung oder eine Version angeben, nicht jedoch beides. Falls keine Version oder Beschriftung für eine bestimmte Datei angegeben wird, wird die letzte Version verwendet.

Content-Repository-Operationen

advanceSearch-Operation

Sucht auf Basis verschiedener Parameter nach Dateien und Ordnern im Repository.

Syntax

```
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
```

Erläuterung:

Beispiel

```
python CADSTool.py advanceSearch --label "Production" --keyword "Quarterly"  
--useDefault -z
```

```
python CADSTool.py advanceSearch --createdStartDate "2009-12-01T00:00:00+01:00"  
--createdEndDate "2010-12-15T21:33:40+01:00" -z
```

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Nach erfolgreichem Abschluss der API wird eine Liste mit allen Dateien und Ordnern angezeigt, die den Suchkriterien entsprechen. Dazu gehören in der Regel die Dateinamen mit ihrem vollständig qualifizierten Pfad und ihren Versionen.
- Error searching files and folders
- error: no such option:<option>

applySecurity-Operation

Legt die Sicherheits-ACL (Access Control List - Zugriffskontrollliste) für eine Datei bzw. einen Ordner im Repository fest.

Syntax

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

Erläuterung:

- *<source>* ist der vollständig qualifizierte Pfad von IBM SPSS Collaboration and Deployment Services Repository für die Datei oder den Ordner, auf den die Sicherheits-ACL angewendet werden soll. Dies ist ein erforderlicher Parameter.
- *<principal>* ist der Benutzer (z. B. *admin*), der als Teil der ACL auf die angegebene Datei oder den angegebenen Ordner angewendet werden soll. Dies ist ein erforderlicher Parameter.
- *<permission>* ist der Berechtigungstyp, der auf die angegebene Datei oder den angegebenen Ordner angewendet werden soll (wie Lesen, Schreiben, Ändern, Löschen oder Besitzer). Dies ist ein erforderlicher Parameter.
- *<provider>* ist der Sicherheitsprovider, der für den Abruf von Informationen zu den Benutzern (Principals) verwendet werden soll. Dies ist ein optionaler Parameter. Zu den gültigen Werten gehören:
 - "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider.
 - AD_<name> für Active Directory, wobei <name> dem Namen des Sicherheitsproviders im System entspricht.
 - ADL_<name> für Active Directory mit lokalem Überschreiben, wobei <name> dem Namen des Sicherheitsproviders im System entspricht.
 - ldap_<name> für OpenLDAP, wobei <name> dem Namen des Sicherheitsproviders im System entspricht.
- `--cascade` wird beim Einstellen der Sicherheit für einen Ordner verwendet, um die Sicherheitseinstellungen auf alle Dateien und Unterordner innerhalb des angegebenen Ordners anzuwenden. Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel werden Sicherheitseinstellungen auf einen Ordner angewendet:

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

Im folgenden Beispiel werden Sicherheitseinstellungen auf einen Ordner und alle darin enthaltenen Dateien und Unterordner angewendet:

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

Im folgenden Beispiel werden Sicherheitseinstellungen auf einen Ordner für einen Principal in einem Active Directory-Sicherheitsprovider namens SPSSAD angewendet:

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

Im folgenden Beispiel werden Sicherheitseinstellungen auf einen Ordner für einen Principal in einem OpenLDAP Directory-Sicherheitsprovider namens LDAP angewendet:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- <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.

cascadeSecurity-Operation

Überträgt die Sicherheitseinstellungen eines Ordners auf alle Dateien und Unterordner innerhalb des Ordners.

Syntax

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

Der Wert von <source> ist der vollständig qualifizierte Pfad des Ordners im Repository. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

copyResource-Operation

Kopiert eine Datei oder einen Ordner in einen anderen Ordner im Repository.

Für diese API gibt es eine Umbenennungsfunktion, über die die angegebene Datei beim Kopieren umbenannt werden kann. Die Fälle, die am Anfang von „[moveResource-Operation](#)“ auf Seite 18 beschrieben werden, gelten auch für diese `copyResource`-API.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Content-Repository-Pfad der Datei/des Ordners, die/der kopiert werden soll. Dies ist ein erforderlicher Parameter.
- `<target>` ist der vollständig qualifizierte Repository-Pfad, in den die Datei/der Ordner kopiert werden soll. Dies ist ein erforderlicher Parameter.

Beispiele

Im folgenden Beispiel wird eine Datei kopiert:

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

Im folgenden Beispiel wird eine Datei kopiert und umbenannt:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- `<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.

createFolder-Operation

Erstellt einen neuen Ordner an einem festgelegten Speicherort im Repository.

Syntax

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

Der Wert von `<source>` ist der vollständig qualifizierte Pfad des neuen Ordners, der erstellt werden soll. Dies ist ein erforderlicher Parameter. Auf Basis des angegebenen Pfads wird der neue Ordner mitsamt etwaiger Unterordner erstellt.

Beispiel

Im folgenden Beispiel wird ein Ordner *Drafts* erstellt, falls er nicht bereits vorhanden ist.

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

- `<source>` Error creating folder.

deleteFile-Operation

Löscht eine Datei sowie all ihre Versionen aus dem Repository.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad der Datei, die gelöscht werden soll. Dies ist ein erforderlicher Parameter.
- `--submittedHierarchy` löscht eine Datei aus dem Ordner "Submitted Jobs". Dies ist ein optionaler Parameter.

Beispiel

Im folgenden Beispiel werden eine Datei sowie all ihre Versionen aus dem Repository gelöscht:

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

Im folgenden Beispiel werden eine Datei sowie all ihre Versionen aus dem Ordner "Submitted Jobs" gelöscht:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

deleteFileVersion-Operation

Löscht eine bestimmte Version einer Datei aus dem Repository.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad der Datei, die gelöscht werden soll. Dies ist ein erforderlicher Parameter.
- `<version>` ist die spezifische Version der Datei, die gelöscht werden soll. Dies ist ein optionaler Parameter.
- `<label>` ist die Beschriftung der Datei, die gelöscht werden soll. Dies ist ein optionaler Parameter.
- `--submittedHierarchy` löscht eine spezifische Version einer Datei aus dem Ordner "Submitted Jobs". Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel wird eine spezifische Version einer Datei gelöscht:

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

Im folgenden Beispiel wird eine Datei mit einer spezifischen Beschriftung gelöscht:

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

Im folgenden Beispiel wird eine Datei mit einer spezifischen Beschriftung aus dem Ordner "Submitted Jobs" gelöscht:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

deleteFolder-Operation

Löscht einen Ordner und seinen gesamten Inhalt aus dem Repository.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad des Ordners, der gelöscht werden soll. Dies ist ein erforderlicher Parameter.
- `--submittedHierarchy` löscht eine spezifische Version des Ordners aus dem Ordner "Submitted Jobs". Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel wird ein Ordner gelöscht:

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

Im folgenden Beispiel wird ein Ordner aus dem Ordner "Submitted Jobs" gelöscht:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

downloadFile-Operation

Lädt eine spezifische Version einer Datei aus dem Repository in das lokale Dateisystem.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad. Dies ist ein erforderlicher Parameter.

- `<version>` ist die Version der Datei, die heruntergeladen werden soll. Dies ist ein optionaler Parameter.
- `<label>` ist die Beschriftung der Datei, die heruntergeladen werden soll. Dies ist ein optionaler Parameter.
- `<target>` ist der vollständig qualifizierte Pfad (im lokalen Dateisystem), in den die Datei heruntergeladen werden soll.

Beispiele

Im folgenden Beispiel wird die letzte Version der Datei heruntergeladen:

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

Im folgenden Beispiel wird mithilfe einer Versionsmarkierung eine spezifische Version der Datei heruntergeladen:

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

Im folgenden Beispiel wird eine beschriftete Version der Datei heruntergeladen:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

export-Operation

Startet einen Export aus dem Content-Repository. Sie können auswählen, welche Dateien und Ordner exportiert werden sollen; die *.pes-Exportdatei wird im lokalen Dateisystem gespeichert.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad des Ordners, der exportiert werden soll. Dies ist ein erforderlicher Parameter.
- `<target>` ist der vollständig qualifizierte Pfad (im lokalen Dateisystem), in dem die *.pes-Exportdatei erstellt werden soll. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

getAccessControllist-Operation

Ruft die Sicherheits-ACL (Access Control List - Zugriffskontrollliste) für eine angegebene Datei/einen angegebenen Ordner im Content-Repository ab.

Syntax

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

Der Wert `<source>` ist der vollständig qualifizierte Pfad der Datei/des Ordners. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

getAllVersions-Operation

Ruft eine Liste aller Versionen einer Datei im Repository ab.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad der Datei, deren Versionen abgerufen werden sollen. Dies ist ein erforderlicher Parameter.
- `--submittedHierarchy` ruft Versionen aus dem Ordner "Submitted Jobs" ab. Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel werden alle Versionen einer angegebenen Datei abgerufen:

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

Im folgenden Beispiel werden alle Versionen einer angegebenen Datei aus dem Ordner "Submitted Jobs" abgerufen:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file versions.
- Wenn der Prozess erfolgreich abgeschlossen wird, werden die Informationen für jede Dateiversion angezeigt, darunter Versionsmarkierung und Beschriftungsinformationen.

getChildren-Operation

Ruft eine Liste aller Dateien und Ordner innerhalb eines angegebenen Repository-Ordners ab.

Syntax

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

Der Wert `<source>` ist der vollständig qualifizierte Pfad des Ordners. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Wenn der Befehl erfolgreich ausgeführt wird, wird der gesamte Inhalt des angegebenen Ordners angezeigt.
- `<source> No such folder exists. Please try again.`
- `<source> Error getting resources.`

getCustomPropertyValue-Operation

Ruft die gültigen Werte ab, die von einer angegebenen benutzerdefinierten Eigenschaft akzeptiert werden.

Syntax

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

Der Wert `<propertyName>` ist der Name der benutzerdefinierten Eigenschaft. Dies ist ein optionaler Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

getMetadata-Operation

Ruft die Metadatenattribute einer Datei oder eines Ordners im Repository ab.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad der Datei oder des Ordners, für die/den Metadaten abgerufen werden sollen. Bei Ordnern werden die Versions-/Beschriftungsattribute ignoriert. Dies ist ein erforderlicher Parameter.
- `<version>` ist die Version der Datei, für die Metadaten abgerufen werden sollen. Dies ist ein optionaler Parameter.
- `<label>` ist die Beschriftung der Datei, für die Metadaten abgerufen werden sollen. Dies ist ein optionaler Parameter.
- `--submittedHierarchy` ruft Metadaten aus dem Ordner "Submitted Jobs" ab. Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel werden Metadaten für einen Ordner abgerufen:

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

Im folgenden Beispiel werden Metadaten für eine beschriftete Version einer Datei oder eines Ordners abgerufen:

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

Im folgenden Beispiel werden Metadaten für eine beschriftete Version einer Datei im Ordner "Submitted Jobs" abgerufen:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file metadata.
- Wenn der Prozess erfolgreich abgeschlossen wird, werden alle Metadateninformationen für die angegebene Datei oder den angegebenen Ordner angezeigt, darunter benutzerdefinierte Metadateneigenschaften.

import-Operation

Importiert eine vorhandene *.pes-Exportdatei aus dem lokalen Dateisystem ins Repository.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Pfad (im lokalen Dateisystem) der *.pes-Exportdatei, die in das Repository importiert werden soll. Dies ist ein erforderlicher Parameter.
- `<target>` ist der vollständig qualifizierte Pfad im Repository, in den die *.pes-Exportdatei importiert werden soll. Dies ist ein erforderlicher Parameter.
- `<type>` gibt den Typ des importierten Inhalts an. Geben Sie einen der folgenden Werte an:
 - **ContentRepository** für Inhaltsobjekte wie Dateien und Ordner.
 - **ResourceDef** für Ressourcendefinitionen
 - **Credential** für Benutzerberechtigungen
 - **DataSource** für Datenquellendefinitionen
 - **MessageDomain** für Nachrichtendomänen
 - **ServerCluster** für Server-Cluster-Definitionen
 - **Server** für Serverdefinitionen
 - **PromotionPolicy** für Hochstufungsrichtlinien

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **ContentRepository** verwendet.

- *<rconflict>* gibt an, wie die doppelte ID oder Namenskonflikte aufgelöst werden sollen. Geben Sie einen der folgenden Werte an:
 - **keepTarget**. Das Zielelement wird beibehalten. Das in der *PES-Datei* enthaltene Quellenobjekt mit der doppelten ID wird ignoriert.
 - **addNewVersion**. Diese Option wird in der Regel verwendet, um ID- oder Namenskonflikte zu beheben. Wenn ein Konflikt mit einer doppelten ID zwischen dem Quellen- und dem Zielobjekt festgestellt wird, wird eine neue Version des Objekts an der Zielposition erstellt. Wenn ein Namenskonflikt festgestellt wird, wird das importierte Objekt an der Zielposition umbenannt. In der Regel wird *_1*, *_2* usw. an die betroffenen Objektnamen angehängt. In dem Fall, dass zwei Versionen eines Objekts dieselbe Beschriftung aufweisen, übernimmt das System eine Beschriftung und verwirft die doppelte Beschriftung, weil zwei Versionen desselben Objekts nicht dieselbe Beschriftung haben dürfen. Die Beschriftung, die beibehalten wird, hängt von dem Parameter **labelFrom** ab.

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **keepTarget** verwendet.

- *<label>* gibt die zu verwendende Beschriftung an, wenn zwei Versionen eines Objekts dieselbe Beschriftung aufweisen. Die Beschriftung für die andere Version wird gelöscht. Geben Sie einen der folgenden Werte an:
 - **source**
 - **target**

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **source** verwendet.

- *<resolution>* definiert die Vorgehensweise bei gesperrten Ressourcen. Geben Sie einen der folgenden Werte an:
 - **continue**. Setzt den Import fort und übergeht alle gesperrten Ressourcen.
 - **abort**. Beendet den Importprozess, wenn gesperrte Ressourcen festgestellt werden. Wenn Konflikte aufgrund von Objektsperren festgestellt werden, wird der Importvorgang nicht fertiggestellt und schlägt fehl.

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **abort** verwendet.

- *<vconflict>* definiert die Vorgehensweise, wenn während des Importprozesses eine ungültige Version festgestellt wird. Geben Sie einen der folgenden Werte an:
 - **import**. Die ungültige Version wird importiert.
 - **discard**. Die ungültige Version wird gelöscht.

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **import** verwendet.

- *<rdefinition>* definiert das Verarbeitungsverhalten für Ressourcendefinitionen. Geben Sie einen der folgenden Werte an:
 - **recommended**. Eine Ressourcendefinition wird nur importiert, wenn die ID oder der Name nicht mit einer Zieldefinition in Konflikt steht. Ressourcendefinitionen, bei denen ein Konflikt besteht, werden nicht importiert.
 - **include**. Es werden sämtliche Ressourcendefinitionen aus der Importdatei importiert. Sie können eine oder mehrere Typen von Ressourcendefinitionen auswählen, die aus dem Import ausgeschlossen werden sollen, indem Sie das entsprechende Kontrollkästchen aktivieren.
 - **exclude**. Es werden keinerlei Ressourcendefinitionen aus den Importdateien importiert. Importierte Objekte müssen möglicherweise so geändert werden, dass sie auf verfügbare Ressourcendefinitionen verweisen.

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert **recommended** verwendet.

- `<exclusion>` definiert die beim Import ausgeschlossenen Ressourcentypen. Mehrere Werte können in einer durch Semikolon getrennten Liste in beliebiger Reihenfolge kombiniert werden. Geben Sie mindestens einen der folgenden Werte an:
 - **credential** schließt Benutzerberechtigungen aus
 - **customproperty** schließt benutzerdefinierte Eigenschaften für Ressourcenobjekte aus
 - **datasource** schließt Datenquellendefinitionen aus
 - **messagedomain** schließt Nachrichtendomänen aus
 - **notification** schließt Benachrichtigungsdefinitionen aus
 - **servercluster** schließt Server-Cluster-Definitionen aus
 - **server** schließt Serverdefinitionen aus
 - **topic** schließt Themendefinitionen aus

Dies ist ein optionaler Parameter. Wenn dieser Parameter nicht angegeben ist, werden alle Typen in den Import eingeschlossen.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

moveResource-Operation

Verschiebt eine Datei oder einen Ordner in einen anderen Ordner im Repository.

Für diese API gibt es eine Umbenennungsfunktion, über die die angegebene Datei/der angegebene Ordner beim Verschieben umbenannt werden kann. Die folgenden Fälle beschreiben das Verhalten der Umbenennungsfunktion:

Die Quelle ist `/Temp Ordner/Temp.txt` und das Ziel ist `/Demoordner`:

- **Fall 1:** Falls der Ordner `Demoordner` vorhanden ist, wird `Temp.txt` in `Demoordner` verschoben.
- **Fall 2:** Falls der Ordner `Demoordner` nicht vorhanden ist, wird `Temp.txt` in `/` verschoben und in `Demoordner` umbenannt.

Falls die Quelle `/Temp Ordner/Temp.txt` und das Ziel `/Demoordner/Abc.dat` ist:

- **Fall 1:** Wenn der Ordner `Demoordner` vorhanden ist, wird `Temp.txt` in den Ordner `Demoordner` verschoben und in `Abc.dat` umbenannt.
- **Fall 2:** Wenn der Ordner `Demoordner` nicht vorhanden ist, wird ein Fehler angezeigt.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Repository-Pfad der Datei/des Ordners, die/der verschoben werden soll. Dies ist ein erforderlicher Parameter.
- `<target>` ist der vollständig qualifizierte Repository-Pfad, in den die Datei/der Ordner verschoben werden soll. Dies ist ein erforderlicher Parameter.

Beispiele

Im folgenden Beispiel wird eine Datei verschoben:

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

Im folgenden Beispiel wird ein Ordner verschoben:

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

Im folgenden Beispiel wird eine Datei verschoben und umbenannt:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- <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.

removeLabel-Operation

Entfernt eine Beschriftung von einer Datei im Repository.

Syntax

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

Erläuterung:

- <source> ist der vollständig qualifizierte Pfad der Datei im Repository. Dies ist ein erforderlicher Parameter.
- <label> ist die Beschriftung der Datei, die von der angegebenen Datei entfernt werden soll. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- 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.

removeSecurity-Operation

Entfernt die Sicherheits-ACL (Access Control List - Zugriffskontrollliste) aus einer angegebenen Datei bzw. einem Ordner im Repository.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Pfad der Datei/des Ordners, von der/dem die Sicherheitseinstellungen entfernt werden sollen. Dies ist ein erforderlicher Parameter.
- `<principal>` ist der Benutzer/Principal (z. B. `admin`), für den die Sicherheitseinstellungen von der angegebenen Datei/dem angegebenen Ordner entfernt werden sollen. Dies ist ein erforderlicher Parameter.
- `<provider>` ist der Sicherheitsprovider, der für den Abruf von Informationen zu den Benutzern (Principals) verwendet werden soll. Dies ist ein optionaler Parameter. Zu den gültigen Werten gehören:
 - "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider.
 - `AD_<name>` für Active Directory, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.
 - `ADL_<name>` für Active Directory mit lokalem Überschreiben, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.
 - `ldap_<name>` für OpenLDAP, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.
- `--cascade` wird beim Entfernen der Sicherheit von einem Ordner verwendet, um die Sicherheitseinstellungen von allen Dateien und Unterordnern innerhalb des angegebenen Ordners zu entfernen. Dies ist ein optionaler Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

search-Operation

Sucht nach Dateien und Ordnern im Repository. Das Ergebnis ist eine Liste der Dateien/Ordner, die den Suchkriterien entsprechen, sowie der Versionen dieser Dateien/Ordner.

Syntax

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

Der Wert von `<criteria>` ist die Suchzeichenfolge, die verwendet wird, um die Metadaten aller Dateien und Ordner im Repository zu durchsuchen. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Nach erfolgreichem Abschluss der Suche wird eine Liste mit allen Dateien und Ordnern angezeigt, die den Suchkriterien entsprechen. Dazu gehören in der Regel die Dateinamen mit ihrem vollständig qualifizierten Pfad und ihren Versionen.
- `<criteria> No file or folder matches the search criteria.`
- `Error searching files and folders.`

setLabel-Operation

Fügt eine Beschriftung zu einer Version einer Datei im Repository hinzu. Falls die Datei bereits beschriftet ist, wird die ursprüngliche Beschriftung entfernt und durch die neue Beschriftung ersetzt.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte Pfad der Datei im Repository. Dies ist ein erforderlicher Parameter.
- `<version>` ist die Version der Datei, der die Beschriftung hinzugefügt werden soll. Dies ist ein erforderlicher Parameter.
- `<label>` ist der Name der Beschriftung, die der angegebenen Version der Datei hinzugefügt werden soll. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

setMetadata-Operation

Fügt Metadateneigenschaften zu Dateien und Ordnern im Repository hinzu.

In Tabelle 4 auf Seite 21 werden die Metadateneigenschaften aufgelistet und es wird angegeben, ob die Metadateneigenschaften Dateien und Ordnern hinzugefügt werden können.

<i>Tabelle 4. Metadateneigenschaften und Ressourcentypen</i>	
Metadateneigenschaft	Ressourcentyp
Autor	Datei
Beschreibung	Datei oder Ordner
Titel	Datei oder Ordner
Ablaufdatum	Datei oder Ordner
Schlüsselwort	Datei
Themen	Datei
Benutzerdefinierte Metadaten	Datei oder Ordner

Syntax

```
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
```

Erläuterung:

- *<source>* ist der vollständig qualifizierte Repository-Pfad der Datei/des Ordners, für die/den Metadaten definiert werden sollen. Dies ist ein erforderlicher Parameter.
- *<author>* ist der Autor der Datei oder des Ordners. Dies ist ein optionaler Parameter.
- *<title>* ist der Titel der Datei oder des Ordners. Dies ist ein optionaler Parameter.
- *<description>* ist die Beschreibung der Datei/des Ordners. Dies ist ein optionaler Parameter.
- *<expirationDate>* ist das Ablaufdatum der Datei oder des Ordners. Dies ist ein optionaler Parameter. Das Datumsformat lautet JJJJ-MM-TTThh:mm:ss:ssZZB (z. B. 1997-07-16T19:20:30+01:00), wobei Folgendes gilt:

JJJJ = vierstellige Jahresangabe

MM = zweistellige Monatsangabe (01 ist Januar usw.)

TT = zweistellige Angabe des Tages (01 bis 31)

hh = zweistellige Angabe der Stunde (00 bis 23)

mm = zweistellige Angabe der Minute (00 bis 59)

ss = zweistellige Angabe der Sekunde (00 bis 59)

ZZB = Zeitzonebezeichnung (Z oder +hh:mm oder -hh:mm)

- *<keyword>* ist das Schlüsselwort für die Datei oder den Ordner. Dies ist ein optionaler Parameter.
- *<version>* ist die spezifische Version der Datei, der Metadaten hinzugefügt werden sollen. Dies ist ein optionaler Parameter.
- *<label>* ist die beschriftete Version der Datei, der Metadaten hinzugefügt werden sollen. Dies ist ein optionaler Parameter.
- *<topic>* ist das Thema, das auf die Datei oder den Ordner angewendet werden soll. Dies ist ein optionaler Parameter.
- *<customProperty>* sind die benutzerdefinierten Eigenschaftswerte, die auf die Datei oder den Ordner angewendet werden sollen. Dies ist ein optionaler Parameter. Die Werte werden als *<customProperty>=<value>* angegeben. Um mehrere benutzerdefinierte Eigenschaften anzuwenden, verwenden Sie ein Semikolon (;) als Trennzeichen (*<customProperty>=<value>;<customProperty>=<value>*). Trennen Sie Eigenschaftswerte mit Mehrfachauswahl durch den Operator | (*<customProperty>=opt1|opt2;<customProperty>=value*).

Anmerkung: Für die Verwendung der setMetadata-API muss mindestens ein optionaler Parameter angegeben werden.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

uploadFile-Operation

Speichert eine Datei aus dem lokalen Dateisystem im Content-Repository, wobei die Option besteht, eine neue Version der Datei zu erstellen, falls die Datei bereits vorhanden ist.

Syntax

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

Erläuterung:

- <source> ist der vollständig qualifizierte Pfad (im lokalen Dateisystem) der Datei, die hochgeladen werden soll. Dies ist ein erforderlicher Parameter.
- <target> ist der vollständig qualifizierte Pfad des Ordners im Repository, in den die Datei hochgeladen werden soll. Dies ist ein erforderlicher Parameter.
- --createVersion gibt an, dass die angegebene Datei bereits vorhanden ist und eine neue Version erstellt werden sollte. Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel ist das Ziel (target) ein vollständig qualifizierter Pfad für *Drafts*:

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

Wenn *MyReport.rptdesign* bereits im Ordner */Demo/Drafts* vorhanden ist, verwenden Sie den Parameter `--createVersion`:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

Prozessmanagementfunktionen

Prozessmanagement-Scripts bieten die Möglichkeit, Jobs zu verwenden. Dieser Bereich enthält folgende Funktionen:

- Ausführen von Jobs
- Abrufen von Jobverläufen
- Abrufen von Jobdetails

In diesem Abschnitt wird die Nutzung von Scripts für Prozessmanagementfunktionen über die Python-Befehlszeile beschrieben. Jede API enthält detaillierte Syntaxinformationen, ein Beispiel und erwartete Nachrichten.

Schlüsselwörter

In [Tabelle 5 auf Seite 24](#) sind die für Prozessmanagement-APIs unterstützten Schlüsselwörter aufgelistet. Die zweite Spalte enthält optionale, abgekürzte Versionen der bereitgestellten Schlüsselwörter. In der Tabelle sind nur Schlüsselwörter aufgelistet, die für Prozessmanagement-APIs spezifisch sind. Zusätzliche Schlüsselwörter, die sowohl für Prozessmanagement-APIs als auch für Repository-APIs gelten, finden Sie in [Tabelle 2 auf Seite 3](#) und [Tabelle 3 auf Seite 5](#).

Schlüsselwort	Optionale Kurzversion	Verwendung
--source	-s	Der Quellenjob mit Pfad
--target	-t	Der Pfad des Zielordners
--notification	-j	Gibt an, dass der Job mit Benachrichtigungen ausgeführt wird
--async	-m	Gibt an, dass der Job asynchron ausgeführt wird
--execId	-y	Die Ausführungs-ID für den Job
--jobStepName	-q	Der Name des Jobschritts
--log		Gibt an, dass Protokolle nicht gelöscht werden sollen. Bei gemeinsamer Verwendung mit --target werden Protokolle an einem Ort gespeichert, der durch --target angegeben wird. Andernfalls werden Protokolle inline angezeigt.

Prozessmanagementoperationen

deleteJobExecutions-Operation

Löscht die angegebenen Jobausführungsobjekte.

Syntax

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

Der Wert von `<execIDs>` ist eine durch Leerzeichen getrennte Liste der IDs für die zu löschenden Ausführungen. Dies ist ein erforderlicher Parameter.

Beispiele

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Execution Id not specified.

executeJob-Operation

Führt einen Job abhängig von den übergebenen Parametern synchron oder asynchron aus. Im Fall einer synchronen Ausführung wird die API erst nach Abschluss des Jobs wieder aktiv. Im Fall einer asynchronen Ausführung wird die API nach Beginn des Jobs wieder aktiv.

Syntax

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

Erläuterung:

- *<source>* ist der vollständig qualifizierte Pfad des Jobs im Repository. Dies ist ein erforderlicher Parameter.
- `--notification` wird verwendet, um den Job mit Benachrichtigungen auszuführen. Dies ist ein optionaler Parameter.
- `--async` wird verwendet, um den Job asynchron auszuführen. Dies ist ein optionaler Parameter.

Beispiele

Im folgenden Beispiel wird der Job synchron und ohne Benachrichtigungen ausgeführt:

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

Im folgenden Beispiel wird der Job synchron und mit Benachrichtigungen ausgeführt:

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

Im folgenden Beispiel wird der Job asynchron und ohne Benachrichtigungen ausgeführt:

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

Im folgenden Beispiel wird der Job asynchron und mit Benachrichtigungen ausgeführt:

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

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

getJobExecutionDetails-Operation

Listet Ausführungsdetails für einen spezifischen Job auf, darunter Jobschritte und Iterationen.

Syntax

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

Erläuterung:

- *<execId>* ist die Ausführungs-ID des Jobs. Dies ist ein erforderlicher Parameter.
- `--log` gibt an, dass das Jobprotokoll inline angezeigt werden soll. Wenn der Parameter `--log` nicht enthalten ist, werden durch den Jobschritt generierte Protokolle nicht angezeigt. Dies ist ein optionaler Parameter.
- *<target>* ist der Ort (im lokalen Dateisystem), an dem die Protokolle gespeichert werden sollen. Dies ist ein optionaler Parameter, und er wird nur zusammen mit dem Parameter `--log` verwendet.

Beispiele

Im folgenden Beispiel werden die Details einer spezifischen Jobausführung aufgelistet:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87b48400" -z
```

Im folgenden Beispiel werden die Details einer spezifischen Jobausführung aufgelistet, wobei das Protokoll inline angezeigt wird:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87b48400" --log -z
```

Im folgenden Beispiel werden die Details einer spezifischen Jobausführung aufgelistet, wobei das Protokoll an einem angegebenen Ort gespeichert wird:

```
python CADSTool.py getJobExecutionDetails --execId "0a58c3710016a7860000010d1a6a87b48400" --log --target "c:\logs" -z
```

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Bei einer erfolgreichen Ausführung werden alle Ausführungsdetails für den Job, Jobschritte und Jobschritt-Iterationen aufgelistet. Protokolle werden inline angezeigt oder an einem bestimmten Ort im lokalen Dateisystem gespeichert.
- `<execId>` No such execution exists. Please try again.
- `<execId>` Error displaying details of a job execution.
- `--target` cannot be used without `--log` parameter

getJobExecutionList-Operation

Listet die momentanen Ausführungen und die abgeschlossenen Ausführungen für einen spezifischen Job und für alle Versionen des Jobs auf.

Syntax

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

Der Wert von `<source>` ist der vollständig qualifizierte Pfad des Jobs im Repository. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

Die folgenden Nachrichten können bei der Verwendung dieser API angezeigt werden:

- Bei einer erfolgreichen Ausführung des angegebenen Jobs werden alle Ausführungsdetails wie Ausführungs-ID, Jobname, Jobausführungsstatus sowie die Beginn- und Endzeit der Jobausführung ausgegeben.
- `<source>` No such job exists. Please try again.
- `<source>` Error displaying execution list for a job.

Kapitel 3. PESImpl-Modul

Die Funktion "IBM SPSS Collaboration and Deployment Services - Essentials for Python" ermöglicht die Interaktion mit Objekten von IBM SPSS Collaboration and Deployment Services Repository direkt innerhalb von Python-Scripts.

Importieren Sie innerhalb Ihres Python-Codes die Klasse `PESImpl` aus dem `pes.api.PESImpl`-Modul. Erstellen Sie ein `PESImpl`-Objekt, das die Verbindungsinformationen für das Repository verwendet, zu dem die Verbindung hergestellt werden soll.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("Benutzer", "Kennwort", "Host", "Port", ssl=True)
```

Die Parameter für den `PESImpl`-Konstruktor lauten wie folgt:

- *Benutzer* entspricht dem Benutzernamen. Der Wert muss ein Präfix enthalten, das den Sicherheitsprovider angibt, wenn es sich nicht um einen Benutzer des nativen Providers handelt. Die folgenden Präfixwerte sind gültig:
 - "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider.
 - `AD_<name>` für Active Directory, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.
 - `ADL_<name>` für Active Directory mit lokalem Überschreiben, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.
 - `ldap_<name>` für OpenLDAP, wobei `<name>` dem Namen des Sicherheitsproviders im System entspricht.

Fügen Sie nach dem Präfix einen Schrägstrich und den Benutzernamen ein. Schließen Sie für Active Directory-Provider die Domäne in das Präfix ein. Für den Benutzer *icrod* in der Domäne *MYDOMAIN* der Active Directory-Instanz *AD_SPSSAD* lautet der Wert beispielsweise *AD_SPSSAD/MYDOMAIN/icrod*. Wenn der Benutzer *icrod* in der Domäne *SPSSLDAP* des OpenLDAP-Providers befindet, lautet der Wert *ldap_SPSSLDAP/icrod*.

- *Kennwort* entspricht dem Kennwort, das dem angegebenen Benutzer zugeordnet wurde.
- *Host* gibt den Namen des Repository-Servers an.
- *Port* gibt die Portnummer für den Repository-Server an.
- `ssl=True` gibt an, dass der Repository-Server das SSL-Protokoll (SSL - Secure Socket Layer) zur Verschlüsselung der Kommunikation verwendet. Wenn der Parameter `ssl` auf `False` gesetzt ist oder wenn der Parameter beim Erstellen des `PESImpl`-Objekts weggelassen wird, wird für die Serverkommunikation nicht SSL verwendet. Bei der Verwendung von SSL muss der Repository-Server für SSL konfiguriert sein. Weitere Informationen finden Sie in der Administratordokumentation.

Alternativ können Sie anstelle der Parameter *Host*, *Port* und *SSL* die Server-URL angeben.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("Benutzer", "Kennwort", server_url="URL")
```

Der Parameter *server_url* gibt die vollständige URL für den Repository-Server an. Verwenden Sie diesen Parameter, wenn Ihr Server einen benutzerdefinierten Kontextpfad verwendet. Der folgende Konstruktor entspricht beispielsweise dem Server `meinServer`, der SSL an Port 443 verwendet und den Kontextpfad `/ibm/spss` aufweist:

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("meinBenutzer", "meinKennwort", server_url="https://meinServer:443/ibm/spss")
```

Anmerkung: Eine IPv6-Adresse muss in eckige Klammern eingeschlossen werden, z. B. `[3ffe:2a00:100:7031::1]`.

Der Zugriff auf bestimmte Methoden erfolgt über das Objekt `pesImpl`.

Content-Repository-API

Content-Repository-Scripts bieten die Möglichkeit, Repository-Ressourcen wie Dateien und Ordner zu verwenden. Dieser Bereich enthält folgende Funktionen:

- Erstellen und Löschen von Ordnern
- Hoch- und Herunterladen von Dateien
- Exportieren und Importieren von Ordnern
- Verwalten von Beschriftungen, Sicherheit und Metadaten

In diesem Abschnitt wird die PESImpl-API beschrieben, die für die Arbeit mit Ressourcen verwendet wird, die im Repository gespeichert sind. Jede Methode enthält detaillierte Syntaxinformationen, ein Beispiel und erwartete Nachrichten.

Methoden

Im folgenden Abschnitt sind alle Content-Repository-Methoden aufgeführt, die für IBM SPSS Collaboration and Deployment Services unterstützt werden.

Anmerkung:

- Verwenden Sie für alle Methoden mit den optionalen Parametern `Label` und `Version` entweder `Label` oder `Version`, nicht jedoch beide Parameter. Falls weder `Version` noch `Label` für eine bestimmte Datei bzw. einen bestimmten Ordner angegeben wird, wird die letzte Version verwendet.
- Für alle Methoden, die einen Pfad zu Dateien/Ordnern im Repository benötigen, kann entweder der Pfad oder der Objekt-URI verwendet werden. Der Objekt-URI ist in den Objekteigenschaften in IBM SPSS Deployment Manager aufgeführt.
- Für Methoden, die die Eingabe von Pfaden für das Quellen- oder Zielrepository oder das Dateisystem erfordern, welche nicht lateinische Unicode-Zeichen enthalten, müssen die Zeichenfolgen als Unicode-Objekte angegeben werden, zum Beispiel:

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

advanceSearch-Methode

Sucht auf Basis verschiedener Parameter, die als Eingabe übergeben werden, nach Dateien und Ordnern im Repository.

Sie können anhand folgender Elemente suchen:

- Autor
- Beschreibung
- Titel
- Erstellt von
- Geändert von
- Ablaufanfangsdatum
- Ablaufenddatum
- MIME-Typ
- Beschriftung
- Schlüsselwort
- Themen
- Erstellungsanfangsdatum
- Erstellungsenddatum
- Anfangsdatum für Versionsänderung

- Enddatum für Objektänderung
- Anfangsdatum für Objektänderung
- Anfangsdatum für Versionsänderung
- Enddatum für Versionsänderung
- URI des übergeordneten Ordners
- Ressourcen-URI

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

Tabelle 6. Eingabeparameter für "advanceSearch"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
criteriaDict	Erforderlich	Wörterbuch	<p>Das Wörterbuch enthält den Schlüssel/Wert des Paares, anhand dessen die Suche ausgeführt wird. Zulässige Schlüsselwerte:</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": "ad□ min", "ti□ tle": "search", "label": "label 1"} </pre>

Tabelle 6. Eingabeparameter für "advanceSearch" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
submittedHierarchy	Optional	Boolesch	Gibt an, ob im Ordner Submitted Jobs gesucht werden soll.	True oder False

Beachten Sie, dass expirationStartDate und expirationEndDate derzeit nicht funktionieren, wenn sie zusammen mit anderen Suchfeldern (wie beispielsweise "title" oder "author") verwendet werden.

Tabelle 7. Rückgabewert für "advanceSearch"

Typ	Beschreibung
PageResult	Struktur, in der jede Zeile einer Suchübereinstimmung entspricht. Weitere Informationen finden Sie im Thema „Klasse "PageResult"“ auf Seite 59.

Tabelle 8. Ausnahmen für "advanceSearch"

Typ	Beschreibung
InsufficientParameterException	Obligatorische Parameter sind nicht angegeben.

Beispiel: Suche nach Beschriftung und Schlüsselwort

Beim folgenden Beispielcode werden alle Dateiversionen mit der Beschriftung *Production* zurückgegeben, die den Schlüsselwortwert *Quarterly* enthalten.

```
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()
```

Beispiel: Suche nach URI

Der folgende Beispielcode gibt alle Dateiversionen der Datei mit dem angegebenen URI zurück.

```
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()
```

applySecurity-Methode

Legt die Sicherheits-ACL (Access Control List - Zugriffskontrollliste) für eine Datei bzw. einen Ordner im Repository fest.

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

Tabelle 9. Eingabeparameter für "applySecurity"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"
<i>principal</i>	Erforderlich	Zeichenfolge	Der Benutzer (z. B. <i>admin</i>), der auf die angegebene Datei bzw. den angegebenen Ordner als Teil der ACL angewendet wird	admin
<i>permission</i>	Erforderlich	Zeichenfolge	Der Berechtigungstyp, der auf die angegebene Datei bzw. den angegebenen Ordner angewendet wird	READ, WRITE, DELETE, MODIFY_ACL ODER OWNER
<i>provider</i>	Optional	Zeichenfolge	Der Sicherheitsprovider, der verwendet wird, um Sicherheitseinstellungen auf Benutzer anzuwenden (z. B. <i>Native</i>). Zu den gültigen Werten gehören: <ul style="list-style-type: none"> • "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider. • AD_<name> für Active Directory, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. • ADL_<name> für Active Directory mit lokalem Überschreiben, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. • ldap_<name> für OpenLDAP, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. 	Native
<i>cascade</i>	Optional	Boolesch	Überträgt die Sicherheitseinstellungen auf alle Dateien und Unterordner im jeweiligen Ordner	True oder False

Tabelle 10. Rückgabewert für "applySecurity"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 11. Ausnahmen für "applySecurity"

Typ	Beschreibung
ResourceNotFoundException	Die Quelldatei ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Der angegebene Name des Benutzers bzw. Sicherheitsproviders ist nicht korrekt.

Beispiel

Im folgenden Beispiel wird einem Benutzer die Berechtigung *READ* (Lesen) für die angegebene Datei erteilt.

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

cascadeSecurity-Methode

Überträgt die Sicherheitseinstellungen eines Ordners auf alle Dateien und Unterordner innerhalb des Ordners.

```
cascadeSecurity(source)
```

Tabelle 12. Eingabeparameter für "cascadeSecurity"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners im Repository	"/Temp Folder" oder "0a58c3670016a7860000010dcee0eaa28219"

Tabelle 13. Rückgabewert für "cascadeSecurity"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 14. Ausnahmen für "cascadeSecurity"

Typ	Beschreibung
ResourceNotFoundException	Der Quellenordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im folgenden Beispiel wird die Sicherheit für den Ordner *Projects* an den Inhalt des Ordners weitergegeben.

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

copyResource-Methode

Kopiert eine Datei oder einen Ordner in einen anderen Ordner im Repository. Die angegebene Quelldatei bzw. der angegebene Quellenordner kann beim Kopieren umbenannt werden.

Weitere Informationen zum Umbenennen finden Sie unter „[moveResource-Methode](#)“ auf Seite 50.

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

Tabelle 15. Eingabeparameter für "copyResource"				
Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners, in den die Datei kopiert wird. Es kann auch ein neuer Dateiname angegeben werden, um die jeweilige Datei bzw. den jeweiligen Ordner beim Kopieren umbenennen.	"/New Folder" oder "/New Folder/abc.dat"

Tabelle 16. Rückgabewert für "copyResource"	
Typ	Beschreibung
Zeichenfolge	URI der kopierten Datei bzw. des kopierten Ordners

Tabelle 17. Ausnahmen für "copyResource"	
Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Der folgende Beispielcode kopiert den Ordner *Drafts* in einen Ordner mit dem Namen *Projects*.

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

createFolder-Methode

Erstellt einen neuen Ordner an einem festgelegten Speicherort im Repository.

```
createFolder(source)
```

Tabelle 18. Eingabeparameter für "createFolder"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der Ordner, der im Repository erstellt werden soll	/Temp Ordner/Beispielordner

Tabelle 19. Rückgabewert für "createFolder"

Typ	Beschreibung
Zeichenfolge	URI des erstellten Ordners

Tabelle 20. Ausnahmen für "createFolder"

Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
ResourceAlreadyExistsException	Der angegebene Ordner ist bereits im Repository vorhanden.

Beispiel

Das folgende Beispiel erstellt einen Ordner mit dem Namen *Drafts* als untergeordnetes Element des Ordners *Demo*. Wenn ein Problem beim Erstellen des Ordners auftritt, wird eine Nachricht über eine Ausnahme an die Konsole gesendet.

```
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."
```

deleteFile-Methode

Löscht eine Datei aus dem Repository. Alle Versionen der Datei werden gelöscht.

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

Tabelle 21. Eingabeparameter für "deleteFile"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a786 0000 010dcee0eaa28219"
submittedHierarchy	Optional	Boolesch	Zeigt an, ob die Datei sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 22. Rückgabewert für "deleteFile"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 23. Ausnahmen für "deleteFile"

Typ	Beschreibung
ResourceNotFoundException	Die Quellendatei ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Die für die Löschung angegebene Ressource ist ein Ordner.

Beispiel

Das folgende Beispiel löscht die Datei *MyReport.rptdesign* aus dem 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."
    
```

deleteFileVersion-Methode

Löscht eine bestimmte Version einer Datei aus dem Repository.

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

Tabelle 24. Eingabeparameter für "deleteFileVersion"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a786000010dcee0eaa28219"
<i>version</i>	Optional. Es muss jedoch entweder <i>version</i> oder <i>label</i> angegeben werden.	Zeichenfolge	Die spezifische Version der Datei, die gelöscht werden soll.	"0:2006-08-25 21:15:49.453"

Tabelle 24. Eingabeparameter für "deleteFileVersion" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>label</i>	Optional. Es muss jedoch entweder <i>version</i> oder <i>label</i> angegeben werden.	Zeichenfolge	Die spezifisch beschriftete Version der Datei, die gelöscht werden soll.	"Version 1"
<i>submittedHierarchy</i>	Optional	Boolesch	Zeigt an, ob die Datei sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 25. Rückgabewert für "deleteFileVersion"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 26. Ausnahmen für "deleteFileVersion"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Die für die Löschung angegebene Ressource ist ein Ordner.

Beispiel

Das folgende Beispiel löscht die Version der Datei *MyReport.rptdesign*, die die Beschriftung *Test* trägt, aus dem Repository.

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

deleteFolder-Methode

Löscht einen Ordner und seine Inhalte aus dem Repository.

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

Tabelle 27. Eingabeparameter für "deleteFolder"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners im Repository	"/Temp Folder" oder "0a58c3670016a786000010dcee0eaa28219"

Tabelle 27. Eingabeparameter für "deleteFolder" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>submittedHierarchy</i>	Optional	Boolesch	Zeigt an, ob der Ordner sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 28. Rückgabewert für "deleteFolder"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 29. Ausnahmen für "deleteFolder"

Typ	Beschreibung
<code>ResourceNotFoundException</code>	Der angegebene Ordner ist nicht vorhanden.
<code>InsufficientParameterException</code>	Erforderliche Parameter sind nicht angegeben.
<code>IllegalParameterException</code>	Die für die Löschung angegebene Ressource ist kein Ordner.

Beispiel

Das folgende Beispiel löscht den Ordner *Drafts* aus dem Repository. Wenn ein Problem beim Löschen des Ordners auftritt, wird eine Nachricht über eine Ausnahme an die Konsole gesendet.

```

from pes.util.PESExceptions 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."
    
```

downloadFile-Methode

Lädt eine spezifische Version einer Datei aus dem Repository in das lokale Dateisystem.

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

Tabelle 30. Eingabeparameter für "downloadFile"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Repository-Pfad der herunterzuladenden Datei	"/Temp Folder/Temp.txt"
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) des Ordners, in den die Datei heruntergeladen werden soll	"C:\Temp"

Tabelle 30. Eingabeparameter für "downloadFile" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>version</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die spezifische Version der Datei, die heruntergeladen werden soll.	"0:2006-08-25 21:15:49.453"
<i>label</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die spezifische beschriftete Version der Datei, die heruntergeladen werden soll.	"Version 2"

Tabelle 31. Rückgabewert für "downloadFile"

Typ	Beschreibung
Resource	Container für Informationen zu einem Repository-Objekt. Weitere Informationen finden Sie im Thema „Klasse "Resource"“ auf Seite 58. .

Tabelle 32. Ausnahmen für "downloadFile"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Mit dem folgenden Beispielcode wird eine mit der Beschriftung *Production* versehene Version der Datei *MyReport.rptdesign* in das Verzeichnis *Shared* im lokalen Dateisystem heruntergeladen.

```
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")
```

exportResource-Methode

Exportiert einen angegebenen Repository-Ordner in eine angegebene *.pes-Exportdatei im lokalen Dateisystem.

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

Tabelle 33. Eingabeparameter für "exportResource"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Repository-Pfad oder Objekt-URI des zu exportierenden Ordners	/Temp Ordner oder 0a58c3670016a78 60000010dcee0eaa28219

Tabelle 33. Eingabeparameter für "exportResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) und Dateiname, in den der Ordner exportiert werden soll.	C:\Temp\backup.pes

Tabelle 34. Rückgabewert für "exportResource"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 35. Ausnahmen für "exportResource"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Bei dem angegebenen Ziel handelt es sich um einen Ordner. Das Ziel muss eine *.pes-Datei sein.

Beispiel

Mit dem folgenden Beispielcode wird der Inhalte des Ordners *Drafts* in eine Exportdatei im Ordner *backups* des lokalen Dateisystems exportiert.

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

getAccessControlList-Methode

Ruft die Sicherheits-ACL (Access Control List = Zugriffskontrollliste) von einer angegebenen Datei bzw. einem Ordner im Repository ab.

```
getAccessControlList(source, submittedHierarchy)
```

Tabelle 36. Eingabeparameter für "getAccessControlList"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a78600 00010dcee0eaa28219"
<i>submittedHierarchy</i>	Optional	Boolesch	Gibt an, ob die Datei bzw. der Ordner sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 37. Rückgabewert für "getAccessControllist"

Typ	Beschreibung
Dictionary	Es wird ein Wörterbuch angezeigt, das den/die Benutzernamen und die zugehörige Berechtigung enthält. Zum Beispiel: {"admin": "MODIFY_ACL", "Joe": "DELETE"}

Tabelle 38. Ausnahmen für "getAccessControllist"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Das folgende Beispiel druckt die ACL für die Datei *MyReport.rptdesign*.

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

getAllVersions-Methode

Ruft eine Liste aller Versionen einer Datei im Repository ab.

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

Tabelle 39. Eingabeparameter für "getAllVersions"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a78600 00010dcee0eaa28219"
submittedHierarchy	Optional	Boolesch	Zeigt an, ob die Datei sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 40. Rückgabewert für "getAllVersions"

Typ	Beschreibung
List	Eine Liste von Ressourcenobjekten. Siehe „Klasse "Resource"“ auf Seite 58.

Tabelle 41. Ausnahmen für "getAllVersions"

Typ	Beschreibung
ResourceNotFoundException	Die Quellendatei ist nicht vorhanden.

Tabelle 41. Ausnahmen für "getAllVersions" (Forts.)

Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Bei der angegebenen Quelle handelt es sich um einen Ordner.

Beispiel

Bei diesem Beispiel werden Informationen zu allen Versionen der Datei *MyReport.rptdesign* abgerufen und jeweils Autor, Versionsmarkierung und Versionsbeschriftung ausgegeben.

```
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()
```

getChildren-Methode

Ruft eine Liste aller Dateien und Ordner innerhalb eines angegebenen Repository-Ordners ab.

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

Tabelle 42. Eingabeparameter für "getChildren"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners im Repository	"/Temp Folder" oder "0a58c3670016a7860000010dcee0eaa28219"
submittedHierarchy	Optional	Boolesch	Zeigt an, ob der Ordner sich im Ordner "Submitted Jobs" befindet	True oder False

Tabelle 43. Rückgabewert für "getChildren"

Typ	Beschreibung
List	Eine Liste von Ressourcenobjekten. Siehe „Klasse "Resource"“ auf Seite 58.

Tabelle 44. Ausnahmen für "getChildren"

Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
ResourceNotFoundException	Der Ordner ist nicht vorhanden.

Beispiel

Mit folgendem Beispielcode wird der Inhalt des Ordners */Demo/Drafts* abgerufen und Titel, Autor und Ressourcen-ID werden ausgegeben.

```
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()
```

getCustomPropertyValue-Methode

Ruft die gültigen Werte ab, die von einer angegebenen benutzerdefinierten Eigenschaft akzeptiert werden.

```
getCustomPropertyValue(propertyName)
```

Tabelle 45. Eingabeparameter für "getCustomPropertyValue"				
Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>propertyName</i>	Erforderlich	Zeichenfolge	Der Name der benutzerdefinierten Eigenschaft.	"FreeForm"

Tabelle 46. Rückgabewert für "getCustomPropertyValue"	
Typ	Beschreibung
List	Gibt einer Liste gültiger Werte zurück, die von der benutzerdefinierte Eigenschaft akzeptiert werden. Falls die Eigenschaft eine Auswahl erfordert (zum Beispiel Einzelauswahl oder Mehrfachauswahl), enthält die Liste alle gültigen Werte für die Auswahl. Falls es eine Freiform-Eigenschaft ist, enthält die Liste den Typ von Daten, der von der Eigenschaft akzeptiert wird (z. B. String, Date oder Number).

Tabelle 47. Ausnahmen für "getCustomPropertyValue"	
Typ	Beschreibung
ResourceNotFoundException	Die angegebene Eigenschaft ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im folgenden Beispiel wird auf die Werte für die benutzerdefinierte Eigenschaft *Language* zugegriffen.

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

getMetadata-Methode

Ruft die Metadatenattribute einer Datei oder eines Ordners im Repository ab, inklusive aller benutzerdefinierten Eigenschaften und Themeninformationen.

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

Tabelle 48. Eingabeparameter für "getMetadata"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a7860 0 00010dcee0eaa28219 "
<i>version</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die spezifische Version der Datei oder des Ordners	"0:2006-08-25 21:15:49.453"
<i>label</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die spezifische beschriftete Version der Datei oder des Ordners	"Version 1"
<i>submittedHierarchy</i>	Optional	Boolesch	Zeigt an, ob die Datei sich im Ordner <i>Submitted Jobs</i> befindet.	True oder False

Tabelle 49. Rückgabewert für "getMetadata"

Typ	Beschreibung
Resource	Container für Informationen zu einem Repository-Objekt. Weitere Informationen finden Sie im Thema „Klasse "Resource"“ auf Seite 58.

Tabelle 50. Ausnahmen für "getMetadata"

Typ	Beschreibung
ResourceNotFoundException	Quellendatei- oder -ordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im folgenden Beispiel wird auf die Ressourcen-ID für den Ordner */Demo/Drafts* zugegriffen.

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


importResource-Methode

Importiert eine vorhandene *.pes-Exportdatei aus dem lokalen Dateisystem ins Repository.

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

Tabelle 51. Eingabeparameter für "importResource"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) der Datei, die importiert werden soll	"C:\Temp\New.pes"
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Repository-Pfad oder Objekt-URI des Ordners, in den der Import erfolgen soll.	"/Temp Folder" oder "0a58c3670016a786000010dcee0eaa28219"
<i>resourceType</i>	Optional	Zeichenfolge	<p>Typ des importierten Inhalts. Geben Sie einen der folgenden Werte an:</p> <ul style="list-style-type: none">• ContentRepository für Inhaltsobjekte wie Dateien und Ordner.• ResourceDef für Ressourcendefinitionen• Credential für Benutzerberechtigungen• DataSource für Datenquellendefinitionen• MessageDomain für Nachrichtendomänen• ServerCluster für Server-Cluster-Definitionen• Server für Serverdefinitionen• PromotionPolicy für Hochstufungsrichtlinien <p>Wenn dieser Parameter nicht angegeben ist, wird der Standardwert ContentRepository verwendet.</p>	

Tabelle 51. Eingabeparameter für "importResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>resourceConflict</i>	Optional	Zeichenfolge	<p>Gibt an, wie die doppelte ID oder Namenskonflikte aufgelöst werden sollen. Geben Sie einen der folgenden Werte an:</p> <ul style="list-style-type: none"> • keepTarget. Das Zielelement wird beibehalten. Das in der <i>PES-Datei</i> enthaltene Quellensobjekt mit der doppelten ID wird ignoriert. • addNewVersion. Diese Option wird in der Regel verwendet, um ID- oder Namenskonflikte zu beheben. Wenn ein Konflikt mit einer doppelten ID zwischen dem Quellen- und dem Zielobjekt festgestellt wird, wird eine neue Version des Objekts an der Zielposition erstellt. Wenn ein Namenskonflikt festgestellt wird, wird das importierte Objekt an der Zielposition umbenannt. In der Regel wird <i>_1</i>, <i>_2</i> usw. an die betroffenen Objektamen angehängt. In dem Fall, dass zwei Versionen eines Objekts dieselbe Beschriftung aufweisen, übernimmt das System eine Beschriftung und verwirft die doppelte Beschriftung, weil zwei Versionen desselben Objekts nicht dieselbe Beschriftung haben dürfen. Die Beschriftung, die beibehalten wird, hängt von dem Parameter labelFrom ab. <p>Wenn dieser Parameter nicht angegeben ist, wird der Standardwert keepTarget verwendet.</p>	
<i>labelFrom</i>	Optional	Zeichenfolge	<p>Zu verwendende Beschriftung, wenn zwei Versionen eines Objekts dieselbe Beschriftung aufweisen. Die Beschriftung für die andere Version wird gelöscht. Geben Sie entweder source oder target an. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert source verwendet.</p>	

Tabelle 51. Eingabeparameter für "importResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>lockResolution</i>	Optional	Zeichenfolge	<p>Definiert die Vorgehensweise bei gesperrten Ressourcen. Geben Sie einen der folgenden Werte an:</p> <ul style="list-style-type: none"> • continue. Setzt den Import fort und übergeht alle gesperrten Ressourcen. • abort. Beendet den Importprozess, wenn gesperrte Ressourcen festgestellt werden. Wenn Konflikte aufgrund von Objektsperren festgestellt werden, wird der Importvorgang nicht fertiggestellt und schlägt fehl. <p>Wenn dieser Parameter nicht angegeben ist, wird der Standardwert abort verwendet.</p>	
<i>invalidVersionConflict</i>	Optional	Zeichenfolge	<p>Definiert die Vorgehensweise, wenn während des Importprozesses eine ungültige Version festgestellt wird. Geben Sie einen der folgenden Werte an:</p> <ul style="list-style-type: none"> • import. Die ungültige Version wird importiert. • discard. Die ungültige Version wird gelöscht. <p>Wenn dieser Parameter nicht angegeben ist, wird der Standardwert import verwendet.</p>	

Tabelle 51. Eingabeparameter für "importResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>resourceDef</i>	Optional	Zeichenfolge	<p>Definiert das Verarbeitungsverhalten für Ressourcendefinitionen. Geben Sie einen der folgenden Werte an:</p> <ul style="list-style-type: none"> • recommended. Eine Ressourcendefinition wird nur importiert, wenn die ID oder der Name nicht mit einer Zieldefinition in Konflikt steht. Ressourcendefinitionen, bei denen ein Konflikt besteht, werden nicht importiert. • include. Es werden sämtliche Ressourcendefinitionen aus der Importdatei importiert. Sie können eine oder mehrere Typen von Ressourcendefinitionen auswählen, die aus dem Import ausgeschlossen werden sollen, indem Sie das entsprechende Kontrollkästchen aktivieren. • exclude. Es werden keinerlei Ressourcendefinitionen aus den Importdateien importiert. Importierte Objekte müssen möglicherweise so geändert werden, dass sie auf verfügbare Ressourcendefinitionen verweisen. <p>Wenn dieser Parameter nicht angegeben ist, wird der Standardwert recommended verwendet.</p>	

Tabelle 51. Eingabeparameter für "importResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>exclude</i>	Optional	Zeichenfolge	<p>Definiert die beim Import ausgeschlossenen Ressourcentypen. Mehrere Werte können in einer durch Semikolon getrennten Liste in beliebiger Reihenfolge kombiniert werden. Geben Sie mindestens einen der folgenden Werte an:</p> <ul style="list-style-type: none"> • credential schließt Benutzerberechtigungen aus • customproperty schließt benutzerdefinierte Eigenschaften für Ressourcenobjekte aus • datasource schließt Datenquellendefinitionen aus • messagedomain schließt Nachrichtendomänen aus • notification schließt Benachrichtigungsdefinitionen aus • servercluster schließt Server-Cluster-Definitionen aus • server schließt Serverdefinitionen aus • topic schließt Themendefinitionen aus <p>Wenn dieser Parameter nicht angegeben ist, werden alle Typen in den Import eingeschlossen.</p>	

Tabelle 52. Rückgabewert für "importResource"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 53. Ausnahmen für "importResource"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im folgenden Beispiel wird der Inhalt der Exportdatei `drafts.pes` in den Ordner `/Demo/Drafts` von IBM SPSS Collaboration and Deployment Services Repository importiert

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

moveResource-Methode

Verschiebt eine Datei oder einen Ordner in einen anderen Ordner im Repository. Eine angegebene Quelldatei kann beim Verschieben umbenannt werden, wobei der Typ und Vorhandensein des Ziels den endgültigen Namen bestimmen.

In der folgenden Tabelle ist das Verhalten der Umbenennungsfunktion beim Verschieben einer Datei beschrieben:

Tabelle 54. Umbenennen von Dateien		
Zieltyp	Zielordner vorhanden	Zielordner nicht vorhanden
Ordner	Quelldatei wird untergeordnetes Element des Zielordners.	Quelldatei wird in den übergeordneten Ordner des angegebenen Ziels verschoben und wird in den Namen des Zielordners umbenannt.
Datei	Quelldatei wird in den Ordner verschoben, der die Zieldatei enthält, und wird in den Namen des Ziels umbenannt.	Fehlernachricht wird ausgegeben.

Wenn es sich bei der Quelle beispielsweise um die Datei `/Temp Ordner/Temp.txt` handelt und das angegebene Ziel der Ordner `/Demoordner` ist, können folgende Ergebnisse auftreten:

- Falls der Ordner `Demoordner` vorhanden ist, wird `Temp.txt` in `Demoordner` verschoben.
- Falls der Ordner `Demoordner` nicht vorhanden ist, wird `Temp.txt` in `/` verschoben und in `Demoordner` umbenannt.

Wenn es sich bei der Quelle um die Datei `/Temp Ordner/Temp.txt` handelt und das angegebene Ziel die Datei `/Demoordner/Abc.dat` ist, können folgende Ergebnisse auftreten:

- Falls der Ordner `Demoordner` vorhanden ist, wird `Temp.txt` in `Demoordner` verschoben und in `Abc.dat` umbenannt.
- Falls der Ordner `Demoordner` nicht vorhanden ist, wird eine Fehlernachricht angezeigt.

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

Tabelle 55. Eingabeparameter für "moveResource"				
Feld	Verwendung	Typ	Beschreibung	Beispielwert
<code>source</code>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	<code>/Temp Folder/Temp.txt</code> oder <code>0a58c3670016a7860000010dcee0eaa28219</code>

Tabelle 55. Eingabeparameter für "moveResource" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
target	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners, in den die Datei verschoben wird. Es kann auch ein neuer Dateiname angegeben werden, um die jeweilige Datei bzw. den jeweiligen Ordner beim Verschieben umzubenennen.	/Neuer Ordner oder /Neuer Ordner/ abc.dat

Tabelle 56. Rückgabewert für "moveResource"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 57. Ausnahmen für "moveResource"

Typ	Beschreibung
ResourceNotFoundException	Die angegebene Quelle ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im folgenden Beispiel wird die Datei MyReport.rptdesign aus dem Ordner /Demo/Drafts in den Ordner /Approved verschoben.

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

removeLabel-Methode

Entfernt eine Beschriftung von einer Datei im Repository.

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

Tabelle 58. Eingabeparameter für "removeLabel"

Feld	Verwendung	Typ	Beispielwert	Beschreibung
source	Erforderlich	Zeichenfolge	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository
label	Erforderlich	Zeichenfolge	"Version 1"	Der Beschriftungsname, der entfernt werden soll

Tabelle 59. Rückgabewert für "removeLabel"

Typ	Beschreibung
Zeichenfolge	URI der aktualisierten Datei

Tabelle 60. Ausnahmen für "removeLabel"

Typ	Beschreibung
ResourceNotFoundException	Die Quellendatei ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Der folgende Beispielcode entfernt die Beschriftung *Draft* aus der Datei *MyReport.rptdesign*.

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

removeSecurity-Methode

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

Entfernt die Sicherheits-ACL (Access Control List - Zugriffskontrollliste) aus einer angegebenen Datei bzw. einem Ordner im Repository.

Tabelle 61. Eingabeparameter für "removeSecurity"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/ Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"
<i>principal</i>	Erforderlich	Zeichenfolge	Der Benutzer (z. B. <i>admin</i>), der die Entfernung aus der angegebenen Datei bzw. dem angegebenen Ordner vornimmt	admin

Tabelle 61. Eingabeparameter für "removeSecurity" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>provider</i>	Optional	Zeichenfolge	<p>Der Sicherheitsprovider (z. B. <i>Native</i>), der verwendet wird, um Informationen zu Benutzern zu erhalten. Die folgenden Werte sind gültig:</p> <ul style="list-style-type: none"> • "Native" für den nativen lokalen systemimmanenten Sicherheitsprovider. Dies ist der Standardprovider. • AD_<name> für Active Directory, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. • ADL_<name> für Active Directory mit lokalem Überschreiben, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. • ldap_<name> für OpenLDAP, wobei <name> dem Namen des Sicherheitsproviders im System entspricht. 	Native
<i>cascade</i>	Optional	Boolesch	Überträgt die Sicherheitseinstellungen auf alle Dateien und Unterordner im jeweiligen Ordner	True oder False

Tabelle 62. Rückgabewert für "removeSecurity"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Tabelle 63. Ausnahmen für "removeSecurity"

Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	Der angegebene Name des Benutzers bzw. Sicherheitsproviders ist nicht korrekt.

Beispiel

Im folgenden Beispiel wird die ACL für einen Principal aus der Datei *MyReport.rptdesign* entfernt.

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

search-Methode

Sucht nach Dateien im Repository und gibt dabei eine Liste der Dateiversionen zurück, die Metadateninhalt aufweisen, der den Suchkriterien entspricht.

```
search(criteria)
```

Tabelle 64. Eingabeparameter für "search"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
criteria	Erforderlich	Zeichenfolge	Der Wert, der für die Suche in den Datei-Metadaten verwendet wird	"Age"

Tabelle 65. Rückgabewert für "search"

Typ	Beschreibung
PageResult	Struktur, in der jede Zeile einer Suchübereinstimmung entspricht. Weitere Informationen finden Sie im Thema „Klasse "PageResult"“ auf Seite 59.

Tabelle 66. Ausnahmen für "search"

Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Im Folgenden wird nach Dateiversionen gesucht, die den Text *Quarterly* in mindestens einem Metadatenfeld enthalten.

```
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()
```

setLabel-Methode

Fügt eine Beschriftung zu einer Version einer Datei im Repository hinzu. Falls die Datei bereits beschriftet ist, wird die ursprüngliche Beschriftung durch die neue Beschriftung ersetzt.

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

Tabelle 67. Eingabeparameter für "setLabel"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"
version	Erforderlich	Zeichenfolge	Die spezifische Version der Datei	"0:2006-08-25 21:15:49.453"

Tabelle 67. Eingabeparameter für "setLabel" (Forts.)

Feld	Verwendung	Typ	Beschreibung	Beispielwert
label	Erforderlich	Zeichenfolge	Die Beschriftung, die zu der Datei hinzugefügt wird	"Version 1"

Tabelle 68. Rückgabewert für "setLabel"

Typ	Beschreibung
Zeichenfolge	URI der aktualisierten Datei

Tabelle 69. Ausnahmen für "setLabel"

Typ	Beschreibung
ResourceNotFoundException	Quellendatei- oder -version ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

Der folgende Beispielcode weist der zweiten Version der Datei *MyReport.rptdesign* die Beschriftung *Beta* zu. Die Methode `getVersionMarker` für ein Objekt vom Typ `Resource` gibt die Markierung zur Beschriftung der Version zurück.

```
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")
```

setMetadata-Methode

Fügt Metadateneigenschaften zu Dateien und Ordnern im Repository hinzu.

In der folgenden Tabelle sind die Metadateneigenschaften aufgeführt und es wird angegeben, ob sie auf Dateien und Ordner angewendet werden können.

Tabelle 70. Eigenschaften von Repository-Objekten

Metadateneigenschaft	Ressourcentyp
Autor	Datei
Beschreibung	Datei oder Ordner
Titel	Datei oder Ordner
Ablaufdatum	Datei oder Ordner
Schlüsselwort	Datei
Themen	Datei
Benutzerdefinierte Metadaten	Datei oder Ordner

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

Tabelle 71. Eingabeparameter für "setMetadata"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000010dcee0eaa28219"
<i>version</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die spezifische Version der Datei, die heruntergeladen werden soll.	"0:2006-08-25 21:15:49.453"
<i>label</i>	Optional. Es kann entweder "version" oder "label" angegeben werden.	Zeichenfolge	Die Beschriftung der spezifischen Version	"Label 1"
<i>props</i>	Erforderlich	Wörterbuch	Enthält alle zu definierenden Metadaten als Schlüssel im Wörterbuch mit dem Metadatenamen. Wie in der Spalte "Beispielwert" gezeigt, wird ein Wert aus topic als Liste und customProperty als Wörterbuch verwendet. Die restlichen Metadaten werden als Zeichenfolge aufgefasst.	<pre>{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a, b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }</pre>

Tabelle 72. Rückgabewert für "setMetadata"

Typ	Beschreibung
Zeichenfolge	URI der Datei bzw. des Ordners, für die/den Metadaten definiert wurden

Tabelle 73. Ausnahmen für "setMetadata"	
Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
ResourceNotFoundException	Quellendatei- oder -ordner ist nicht vorhanden.

Beispiel

Der folgende Beispielcode weist der Version *Production* der Datei *MyReport.rptdesign* das Schlüsselwort *Quarterly* zu.

```
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
```

uploadFile-Methode

Speichert eine Datei aus dem lokalen Dateisystem im Repository, wobei die Option besteht, eine neue Version der Datei zu erstellen, falls die Datei bereits vorhanden ist.

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

Tabelle 74. Eingabeparameter für "uploadFile"				
Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) der Datei, die hochgeladen werden soll	"C:\Temp\Temp.txt"
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad des Zielordners in dem Repository, in den die Datei hochgeladen werden soll	"/Temp Folder"
<i>versionFlag</i>	Optional	Boolesch	Falls die angegebene Datei bereits vorhanden ist, wird eine neue Datei erstellt	True oder False

Tabelle 75. Rückgabewert für "uploadFile"	
Typ	Beschreibung
Zeichenfolge	URI der hochgeladenen Datei

Tabelle 76. Ausnahmen für "uploadFile"	
Typ	Beschreibung
ResourceNotFoundException	Der Quellen- oder Zielordner ist nicht vorhanden.
ResourceAlreadyExistsException	Eine Datei bzw. ein Ordner mit dem Namen der Quelldatei ist im Zielordner vorhanden und der Parameter <i>versionFlag</i> ist nicht angegeben.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.

Beispiel

In diesem Beispiel wird die Datei *MyReport.rptdesign* in den Ordner */Demo/Drafts* im Repository hochgeladen. Falls die Datei bereits vorhanden ist, wird aufgrund der Angabe des Parameters `versionFlag` eine neue Version der Datei hochgeladen.

```
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
```

Wrapperklassen

Die `PESImpl`-API beinhaltet Klassen, die als Wrapper für Objekte fungieren, die von den durch die Content-Repository-Methoden aufgerufenen Web-Services zurückgegeben werden. Diese Wrapperklassen bieten eine Schnittstelle zur Anzeige der von den Methoden zurückgegebenen Informationen.

Klasse "Resource"

Die Klasse `Resource` fungiert als vereinfachter Wrapper für das Repository-Objekt `ResourceSpecifier`. `Resource` und bietet Zugriff auf objektspezifische Informationen.

Neben den Standardmetadaten, die zu Repository-Objekten gehören, enthält diese Klasse auch etwaige benutzerdefinierte Metadaten-Informationen, die für Objekte im Repository definiert wurden. In [Tabelle 77](#) auf Seite 58 sind alle Methoden aufgelistet, die in der Klasse `Resource` verfügbar sind.

Methodenname	Beschreibung
<code>getAccessControlList</code>	Gibt ein Wörterbuch der Sicherheitsberechtigungen eines Objekts zurück. Es enthält den Benutzernamen als Schlüssel und ausschließlich die höchste Berechtigung, über die der Benutzer verfügt. Beispiel: Wenn der Benutzer <i>Joe</i> über die Berechtigung <i>DELETE</i> (Löschen) für die <i>Ressource X</i> verfügt, gibt <code>getAccessControlList</code> für das <i>X</i> nicht alle drei Berechtigungen (Lese-, Schreib- und Löschberechtigung) aus dem Web-Service-Aufruf, sondern <code>{ ' Joe ' : ' DELETE ' }</code> zurück.
<code>getOwner</code>	Gibt den Namen des Besitzers des Objekts als Zeichenfolge zurück.
<code>getAuthor</code>	Gibt den Namen des Autors des Objekts als Zeichenfolge zurück.
<code>getContentSize</code>	Gibt die Größe des Objekts zurück.
<code>getCreatedBy</code>	Gibt den Namen des Benutzers, der das Objekt erstellt hat, als Zeichenfolge zurück.
<code>getCreationDate</code>	Gibt das Erstellungsdatum des Objekts als <code>Datetime</code> -Objekt zurück.
<code>getDescription</code>	Gibt die Beschreibung des Objekts als Liste zurück.
<code>getDescriptionLanguage</code>	Gibt die Sprache des Objekts als Liste zurück.
<code>getExpirationDate</code>	Gibt das Ablaufdatum des Objekts als <code>Datetime</code> -Objekt zurück.
<code>isExpired</code>	Gibt an, ob das angegebene Objekt abgelaufen ist oder nicht
<code>getMIMEType</code>	Gibt den MIME-Typ des Objekts als Zeichenfolge zurück.

Tabelle 77. Methoden für die Klasse "Resource" (Forts.)

Methodenname	Beschreibung
getModificationDate	Gibt das letzte Änderungsdatum des Objekts als Datetime-Objekt zurück.
getObjectCreationDate	Gibt das Objekterstellungsdatum des Objekts als Datetime-Objekt zurück.
getObjectLastModifiedBy	Gibt den Benutzer, der das Objekt zuletzt geändert hat, als Zeichenfolge zurück.
getObjectLastModifiedDate	Gibt das letzte Objekt-Änderungsdatum des Objekts als Datetime-Objekt zurück.
getResourceID	Gibt die Ressourcen-ID des Objekts als Zeichenfolge zurück.
getResourcePath	Gibt den Pfad des angegebenen Objekts als Zeichenfolge zurück.
getTitle	Gibt den Titel für das Objekts als Zeichenfolge zurück.
getTopicList	Gibt die Themenliste für das Objekt zurück.
getVersionMarker	Gibt die Version des Objekts als Zeichenfolge zurück.
getVersionLabel	Gibt die Beschriftung des Objekts als Zeichenfolge zurück.
getCustomMetadata	Gibt benutzerdefinierte Eigenschaften, die mit dem Objekt verknüpft sind, als Wörterbuch zurück.
getKeywordList	Gibt eine Liste von Schlüsselwörtern, die mit dem Objekt verknüpft sind, zurück.

Klasse "IdentificationSpecifier"

Diese Klasse fungiert als vereinfachter Wrapper für das Repository-Objekt `IdentificationSpecifier` und bietet Zugriff auf informationsspezifische Daten für das Objekt.

In [Tabelle 78](#) auf Seite 59 sind alle Methoden aufgelistet, die in der Klasse `IdentificationSpecifier` verfügbar sind.

Tabelle 78. Methoden für die Klasse "IdentificationSpecifier"

Methodenname	Beschreibung
getIdentifizier	Gibt den ID-Wert eines Objekts als Zeichenfolge zurück.
getVersionMarker	Gibt die Version eines Objekts als Zeichenfolge zurück.
getVersionLabel	Gibt die einer Objektversion hinzugefügte Beschriftung als Zeichenfolge zurück.

Klasse "PageResult"

Die Klasse `PageResult` dient als Container für Suchergebnisse. Ein einzelner Treffer in den Ergebnissen entspricht Zeilen im Objekt `PageResult`.

Beispielsweise führt eine Suche, die vier Ressourcen zurückgibt, zu einem Objekt `PageResult` mit vier Zeilen. In [Tabelle 79](#) auf Seite 59 sind alle Methoden aufgelistet, die in der Klasse `PageResult` verfügbar sind.

Tabelle 79. Methoden für die Klasse "PageResult"

Methodenname	Beschreibung
getRows	Gibt eine Liste von <code>SearchRow</code> -Objekten zurück. Weitere Informationen finden Sie im Thema „Klasse "SearchRow"“ auf Seite 60.

Klasse "SearchRow"

Die Klasse SearchRow dient als Container für Informationen auf der Objektebene zu einem einzelnen Suchergebnis. Sie können mithilfe der Methoden dieser Klasse auf Metadaten zu einem Objekt zugreifen.

In [Tabelle 80 auf Seite 60](#) sind alle Methoden aufgelistet, die in der Klasse SearchRow verfügbar sind.

Methodenname	Beschreibung
getTitle	Gibt den Namen der Datei bzw. des Ordners zurück.
getAuthor	Gibt den Autor der Datei bzw. des Ordners zurück.
getMIMEType	Gibt den MIME-Typ der Datei zurück.
getObjectLastModifiedBy	Gibt den Benutzer zurück, der die Datei bzw. den Ordner zuletzt geändert hat.
getModified	Gibt das Datum und die Uhrzeit der letzten Änderung der Datei bzw. des Ordners zurück.
getFolderPath	Gibt den Speicherort der Datei bzw. des Ordners zurück.
getFolder	Gibt den Namen des übergeordneten Ordners der Datei bzw. des Ordners zurück.
getParentURI	Gibt den Objekt-URI des übergeordneten Ordners zurück.
getTopic	Gibt die Themen zurück, die mit der Datei bzw. dem Ordner verknüpft sind.
getChildRow	Gibt die Liste mit SearchChildRow-Objekten zurück (im nächsten Abschnitt finden Sie weitere Informationen).

Um für ein Objekt auf Informationen auf der Versionsebene zuzugreifen, verwenden Sie die Methode getChildRow, um untergeordnete Zeilen auszugeben, die den Objektversionen entsprechen.

Klasse "SearchChildRow"

Die Klasse SearchChildRow dient als Container für Informationen auf der Versionsebene zu einem einzelnen Suchergebnis. Sie können mithilfe der Methoden dieser Klasse auf Metadaten zu einer Objektversion zugreifen.

In [Tabelle 81 auf Seite 60](#) sind alle Methoden aufgelistet, die in der Klasse SearchChildRow verfügbar sind.

Methodenname	Beschreibung
getExpirationDate	Gibt das Ablaufdatum der Datei bzw. des Ordners zurück.
getKeyword	Gibt die Schlüsselwörter zurück, die mit der Version der Datei bzw. des Ordners verknüpft sind.
getVersionLabel	Gibt die Versionsbeschriftung der Datei bzw. des Ordners zurück.
getDescription	Gibt die Beschreibung der Datei bzw. des Ordners zurück.
getLanguage	Gibt die Sprache zurück.
getVersionCreationDate	Gibt das Datum und die Uhrzeit der Erstellung der Datei bzw. des Ordners zurück.
getVersionMarker	Gibt die Versionsmarkierung der Datei bzw. des Ordners zurück.

Tabelle 81. Methoden für die Klasse "SearchChildRow" (Forts.)

Methodenname	Beschreibung
getUri	Gibt den Objekt-URI der Datei bzw. des Ordners zurück.

Prozessmanagement-API

Prozessmanagement-Scripts bieten die Möglichkeit, Jobs zu verwenden. Dieser Bereich enthält folgende Funktionen:

- Ausführen von Jobs
- Abrufen von Jobverläufen
- Abrufen von Jobdetails

In diesem Abschnitt sind die PESImpl-Methoden beschrieben, die für die Arbeit mit Jobs verwendet werden, die im Repository gespeichert sind. Jede Methode enthält detaillierte Syntaxinformationen, ein Beispiel und erwartete Nachrichten.

Methoden

Im folgenden Abschnitt sind alle Prozessmanagement-Scripting-Methoden aufgeführt, die für IBM SPSS Collaboration and Deployment Services unterstützt werden.

Hinweis: Bei allen Methoden, für die ein Pfad zu Dateien/Ordern im Repository erforderlich ist, kann entweder der Pfad oder der Objekt-URI verwendet werden. Der Objekt-URI ist in den Objekteigenschaften in IBM SPSS Deployment Manager aufgeführt.

cancelJob-Methode

Bricht einen laufenden Job ab.

```
cancelJob(executionId)
```

Tabelle 82. Eingabeparameter für "cancelJob"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Zeichenfolge	Ausführungs-ID für den Job	0a58c33d002ce90800 00010e0ccf7b01800e

Tabelle 83. Rückgabewert für "cancelJob"

Typ	Beschreibung
Boolesch	Gibt eine Erfolgsmeldung zurück, wenn der Job abgebrochen wird.

Beispiel

In diesem Beispiel wird die Ausführung des Jobs *Reports* beendet.

```
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
```

deleteJobExecutions-Methode

Löscht eine oder mehrere Jobausführungen.

```
deleteJobExecutions(executionId)
```

Tabelle 84. Eingabeparameter für "deleteJobExecutions"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Liste	Liste der Jobausführungs-IDs	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

Tabelle 85. Rückgabewert für "deleteJobExecutions"

Typ	Beschreibung
Boolesch	True oder False, abhängig davon, ob die API erfolgreich ausgeführt wird.

Beispiel

In diesem Beispiel werden die Ausführungen für den Job *Reports* gelöscht.

```
from pes.util.PESEExceptions 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)
```

executeJob-Methode

Führt einen Job abhängig von den übergebenen Parametern synchron oder asynchron aus. Im Fall einer synchronen Ausführung wird die Methode erst nach Abschluss des Jobs wieder aktiv. Im Fall einer asynchronen Ausführung wird die Methode nach Beginn des Jobs wieder aktiv.

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

Tabelle 86. Eingabeparameter für "executeJob"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) der Datei, die hochgeladen werden soll	"C:\Temp\Temp.txt"
<i>notification</i>	Optional	Boolesch	Gibt an, ob der Job mit oder ohne Benachrichtigungen ausgeführt wird. Die Standardeinstellung ist False.	True oder False
<i>asynchronous</i>	Optional	Boolesch	Gibt an, ob der Job asynchron ausgeführt wird. Die Standardeinstellung ist False.	True oder False

Tabelle 87. Rückgabewert für "executeJob"

Typ	Beschreibung
Zeichenfolge	Die eindeutige Kennung für die Ausführung des Jobs. Diese Kennung wird verwendet, um auf eine bestimmte Jobausführung zu verweisen.

Beispiel

In diesem Beispiel wird die Ausführung des Jobs *Reports* asynchron mit Benachrichtigungen initiiert.

```
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
```

getJobExecutionDetails-Methode

Enthält die Ausführungsdetails für einen spezifischen Job, darunter Jobschritte und Iterationen.

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

Tabelle 88. Eingabeparameter für "getJobExecutionDetails"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Zeichenfolge	Die Ausführungs-ID des Jobs	0a58c33d002ce9080000010e0ccf7b01800e
<i>log</i>	Optional	Boolesch	Gibt an, ob das Jobprotokoll inline angezeigt wird	True oder False
<i>target</i>	Optional	Zeichenfolge	Der Ort (im lokalen Dateisystem), an dem die Protokolle gespeichert werden sollen. Wird nur zusammen mit dem --log-Parameter verwendet.	"C:\logs"

Tabelle 89. Rückgabewert für "getJobExecutionDetails"

Typ	Beschreibung
jobExecutionDetails	Details zu einer Jobausführung. Weitere Informationen finden Sie im Thema „Klasse "jobExecutionDetails"“ auf Seite 65.

Beispiel

In diesem Beispiel werden Informationen zu Ausführungen von Jobschritten für die Jobausführung mit der Kennung *execId* abgerufen, wobei das Ergebnis für jeden Schritt an die Konsole gesendet wird.

```
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()
```

getJobExecutionList-Methode

Enthält die Ausführungen für einen spezifischen Job, inklusive aller momentan ausgeführten und abgeschlossenen Jobs, und für alle Versionen des Jobs.

```
getJobExecutionList(source)
```

Tabelle 90. Eingabeparameter für "JobExecutionList-API"

Feld	Verwendung	Typ	Beschreibung	Beispielwert
source	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad des Jobs im Repository	"/testJob"

Tabelle 91. Rückgabewert für "getJobExecutionList"

Typ	Beschreibung
PageResult	Container für die Liste mit Ausführungen für einen Job. Weitere Informationen finden Sie im Thema „Klasse "PageResult"“ auf Seite 64. .

Beispiel

In diesem Beispiel werden die Ausführungen für den Job *Reports* abgerufen, wobei Informationen zu jeder Ausführung an die Konsole gesendet werden.

```
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()
```

Wrapperklassen

Die PESImpl-API beinhaltet Klassen, die als Wrapper für Objekte fungieren, die von den durch die Prozessmanagementmethoden aufgerufenen Web-Services zurückgegeben werden. Diese Wrapperklassen bieten eine Schnittstelle zur Anzeige der von den Methoden zurückgegebenen Informationen.

Klasse "PageResult"

Diese Klasse `PageResult` dient als Container für Jobausführungsergebnisse, wodurch das Abrufen spezifischer Daten zur Jobausführung möglich ist.

Eine einzelne Jobausführung entspricht Zeilen im Objekt `PageResult`. Ein Job, der beispielsweise viermal ausgeführt wurde, entspricht einem Objekt `PageResult` mit vier Zeilen. In [Tabelle 92 auf Seite 64](#) sind alle Methoden aufgelistet, die in der Klasse `PageResult` verfügbar sind.

Tabelle 92. Methoden für die Klasse "PageResult"

Methodenname	Beschreibung
getRows	Gibt eine Liste mit Zeilenobjekten zurück, von denen eine jede die Ausführung eines Jobs darstellt. Weitere Informationen finden Sie im Thema „Klasse "Row"“ auf Seite 64.

Klasse "Row"

Die Klasse `Row` dient als Container für Informationen zu einer Jobausführung auf Jobebene. Sie können mithilfe der Methoden dieser Klasse auf Metadaten zu einer Jobausführung zugreifen.

In [Tabelle 93 auf Seite 65](#) sind alle Methoden aufgelistet, die in der Klasse `Row` verfügbar sind.

Tabelle 93. Methoden für die Klasse "Row"

Methodenname	Beschreibung
getObjId	Gibt die Ausführungs-ID des Jobs zurück.
getPath	Gibt den Pfad des Jobs zurück.
getVersionMarker	Gibt die Versionsmarkierung des Jobs zurück, der ausgeführt wurde.
getVersionLabel	Gibt die Versionsbeschriftung des Jobs zurück, der ausgeführt wurde.
getEventObjId	Gibt die Ereignis-ID des Jobs zurück, der ausgeführt wurde.
getEventState	Gibt den Status des momentan ausgeführten Jobs zurück.
getEventCompletionCode	Gibt den Fertigstellungscode des Jobs zurück.
getEventStartDateTime	Gibt Anfangsdatum und -zeit des Jobs zurück.
getEventEndDateTime	Gibt Enddatum und Endzeit des Jobs zurück.
getQueuedDateTime	Gibt das Datum und die Zeit der Aufnahme des Jobs in die Warteschlange zurück.

Klasse "jobExecutionDetails"

Diese Klasse wird von der Methode `getJobExecutionDetails` zurückgegeben. Sie speichert die Ausführungsdetails für einen Job und umfasst eine Liste der `jobStepExecution`-Objekte, die Informationen zu jedem Schritt im Job enthalten.

In Tabelle 94 auf Seite 65 sind alle Methoden aufgelistet, die in der Klasse `jobExecutionDetails` verfügbar sind.

Tabelle 94. Methoden für die Klasse "jobExecutionDetails"

Methodenname	Beschreibung
getJobStepDetails	Gibt eine Liste von <code>jobStepExecutionDetails</code> -Objekten zurück. Weitere Informationen finden Sie im Thema „Klasse "jobStepExecutionDetails"“ auf Seite 66.
getArtifactLocation	Gibt eine Liste von Speicherorten von Jobartefakten zurück.
getCompletionCode	Gibt den Fertigstellungscode der Jobausführung zurück.
getEndDateTime	Gibt Enddatum und Endzeit der Jobausführung zurück.
getEventName	Gibt den Ereignisnamen der Jobausführung zurück.
getEventUUID	Gibt die Ereignis-ID der Jobausführung zurück.
getExecutionState	Gibt den Ausführungsstatus der Jobausführung zurück.
getExecutionSuccess	Gibt den Status der Jobausführung hinsichtlich Erfolg/Fehlschlag zurück.

Tabelle 94. Methoden für die Klasse "jobExecutionDetails" (Forts.)

Methodenname	Beschreibung
getExecutionWarning	Gibt an, ob Warnungen gegeben wurden.
getLog	Gibt das generierte Protokoll (als Zeichenfolge) zurück.
getNotificationEnabled	Gibt an, ob E-Mail-Benachrichtigungen aktiviert sind oder nicht.
getQueuedDateTime	Gibt das Datum und die Zeit der Aufnahme der Jobausführung in die Warteschlange zurück.
getStartDateTime	Gibt Anfangsdatum und -zeit der Jobausführung zurück.
getUserName	Gibt den Namen des Benutzers zurück, der den Job ausgeführt hat.
getUUID	Gibt die Ausführungs-ID des Jobs zurück.

Klasse "jobStepExecutionDetails"

Diese Klasse speichert die Ausführungsdetails für einen Jobschritt sowie eine Liste von jobStepChildExecutionDetails-Objekten. Diese Klasse enthält das Objekt ExecutionDetails, an das sie all ihre Methodenaufrufe delegiert.

In Tabelle 95 auf Seite 66 sind alle Methoden aufgelistet, die in der Klasse jobStepExecutionDetails verfügbar sind.

Tabelle 95. Methoden für die Klasse "jobStepExecutionDetails"

Methodenname	Beschreibung
getJobStepChldExecutionList	Gibt eine Liste von jobStepChildExecutionDetails-Objekten zurück. Weitere Informationen finden Sie im Thema „Klasse "jobStepChildExecutionDetails"“ auf Seite 67.
getArtifactLocation	Gibt eine Liste von Speicherorten von Jobschritt-Artefakten zurück.
getCompletionCode	Gibt den Fertigstellungscode des Jobschritts zurück.
getEndDateTime	Gibt Enddatum und Endzeit des Jobschritts zurück.
getEventName	Gibt den Ereignisnamen des Jobschritts zurück.
getEventUUID	Gibt die Ereignis-ID des Jobschritts zurück.
getExecutionState	Gibt den Ausführungsstatus des Jobschritts zurück.
getExecutionSuccess	Gibt den Status des Jobschritts hinsichtlich Erfolg/Fehl-schlag zurück.
getExecutionWarning	Gibt an, ob Warnungen gegeben wurden.
getLog	Gibt das generierte Protokoll (als Zeichenfolge) zurück.
getNotificationEnabled	Gibt an, ob E-Mail-Benachrichtigungen aktiviert sind oder nicht.
getQueuedDateTime	Gibt das Datum und die Zeit der Aufnahme des Jobschritts in die Warteschlange zurück.
getStartDateTime	Gibt Anfangsdatum und -zeit des Jobschritts zurück.

Tabelle 95. Methoden für die Klasse "jobStepExecutionDetails" (Forts.)

Methodenname	Beschreibung
getUserName	Gibt den Namen des Benutzers zurück, der den Jobschritt ausgeführt hat.
getUUID	Gibt die Ausführungs-ID des Jobschritts zurück.

Klasse "jobStepChildExecutionDetails"

Die Klasse `jobStepChildExecutionDetails` dient als Container für untergeordnete Ausführungen einzelner Jobschritte. Ein iterativer Berichtsjobschritt z. B. erstellt eine untergeordnete Ausführung für jede Iteration des Schritts. Sie können mithilfe der Methoden dieser Klasse auf Metadaten zu einer untergeordneten Ausführung zugreifen.

In Tabelle 96 auf Seite 67 sind alle Methoden aufgelistet, die in der Klasse `jobStepChildExecutionDetails` verfügbar sind.

Tabelle 96. Methoden für die Klasse "jobStepChildExecutionDetails"

Methodenname	Beschreibung
getArtifactLocation	Gibt eine Liste von untergeordneten Ausführungen von Jobartefakten zurück.
getCompletionCode	Gibt den Fertigstellungscode der untergeordneten Ausführung zurück.
getEndDateTime	Gibt Enddatum und Endzeit der untergeordneten Ausführung zurück.
getEventName	Gibt den Ereignisnamen der untergeordneten Ausführung zurück.
getEventUUID	Gibt die Ereignis-ID der untergeordneten Ausführung zurück.
getExecutionState	Gibt den Ausführungsstatus der untergeordneten Ausführung zurück.
getExecutionSuccess	Gibt den Status der untergeordneten Ausführung hinsichtlich Erfolg/Fehlschlag zurück.
getExecutionWarning	Gibt an, ob Warnungen gegeben wurden.
getLog	Gibt das generierte Protokoll (als Zeichenfolge) zurück.
getNotificationEnabled	Gibt an, ob E-Mail-Benachrichtigungen aktiviert sind.
getQueuedDateTime	Gibt das Datum und die Zeit der Aufnahme der untergeordneten Ausführung in die Warteschlange zurück.
getStartDateTime	Gibt Anfangsdatum und -zeit der untergeordneten Ausführung zurück.
getUserName	Gibt den Namen des Benutzers zurück, der die untergeordnete Ausführung ausgeführt hat.
getUUID	Gibt die Ausführungs-ID der untergeordneten Ausführung zurück.

Beispielscripts

Beispielscripts, die die Verwendung der Klasse PESImpl veranschaulichen, sind in folgendem Verzeichnis installiert:

`<Installationsposition>/samples`

Diese Scripts führen eine Reihe von Aufgaben durch, unter anderem:

- Löschen abgelaufener Elemente aus IBM SPSS Collaboration and Deployment Services Repository
- Löschen abgelaufener übergebener Artefakte
- Löschen von Jobverläufen

Sie können die Scripts aus einem allgemeinen Jobschritt in IBM SPSS Deployment Manager aufrufen, um Wartungsaufgaben für das Repository durchzuführen.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. IBM stellt dieses Material möglicherweise auch in anderen Sprachen zur Verfügung. Für den Zugriff auf das Material in einer anderen Sprache kann eine Kopie des Produkts oder der Produktversion in der jeweiligen Sprache erforderlich sein.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France*

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

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

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingun-

gen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die angeführten Leistungsdaten und Kundenbeispiele dienen nur zur Illustration. Die tatsächlichen Ergebnisse beim Leistungsverhalten sind abhängig von der jeweiligen Konfiguration und den Betriebsbedingungen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Hinweise zur Datenschutzrichtlinie

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. In vielen Fällen werden von den Softwareangeboten keine personenbezogenen Daten erfasst. Einige der IBM Softwareangebote können Sie jedoch bei der Erfassung personenbezogener Daten unterstützen. Wenn dieses Softwareangebot Cookies zur Erfassung personenbezogener Daten verwendet, sind nachfolgend nähere Informationen über die Verwendung von Cookies durch dieses Angebot zu finden.

Dieses Softwareangebot verwendet keine Cookies oder andere Technologien zur Erfassung personenbezogener Daten.

Wenn es die für dieses Softwareangebot bereitgestellten Konfigurationen Ihnen als Kunde ermöglichen, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, müssen Sie sich zu allen gesetzlichen Bestimmungen in Bezug auf eine solche Datenerfassung, einschließlich aller Mitteilungspflichten und Zustimmungsanforderungen, rechtlich beraten lassen.

Weitere Informationen zur Nutzung verschiedener Technologien, einschließlich Cookies, für diese Zwecke finden Sie in der "IBM Online-Datenschutzerklärung, Schwerpunkte" unter <http://www.ibm.com/privacy>, in der "IBM Online-Datenschutzerklärung" unter <http://www.ibm.com/privacy/details> im Abschnitt "Cookies, Web-Beacons und sonstige Technologien" und in "IBM Software Products and Software-as-a-Service Privacy Statement" unter <http://www.ibm.com/software/info/product-privacy>.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein.

