

IBM SPSS Collaboration and
Deployment Services - Essentials
for Python 4.2-Referenz



Note: Before using this information and the product it supports, read the general information under Notices auf S. 66.

This document contains proprietary information of SPSS Inc, an IBM Company. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

When you send information to IBM or SPSS, you grant IBM and SPSS a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright SPSS Inc. 2004, 2010..**

Vorwort

IBM® SPSS® Collaboration and Deployment Services ist eine Anwendung auf Unternehmensebene, die die weit verbreitete Verwendung von Vorhersageanalysen ermöglicht. IBM SPSS Collaboration and Deployment Services bietet eine zentrale, sichere und Audit-fähige Speicherung von Analyseeinrichtungen, erweiterte Funktionen für die Verwaltung und Steuerung von Analyseprozessen zur Vorhersage sowie ausgereifte Mechanismen zur Bereitstellung der Ergebnisse der analytischen Verarbeitung für die Endbenutzer. Zu den Vorteilen von IBM SPSS Collaboration and Deployment Services gehören die Sicherung des Werts von Analyseeinrichtungen, die Gewährleistung der Einhaltung von Bestimmungen, die Verbesserung der Produktivität der Analysten sowie die Minimierung der IT-Kosten für die Verwaltung von Analytiken.

Technischer Support

Registrierte Kunden von SPSS Inc. können den SPSS Inc. Technischen Support in Anspruch nehmen. Kunden können sich an den technischen Support wenden, wenn sie Hilfe bei der Arbeit mit SPSS Inc.-Produkten oder bei der Installation in einer der unterstützten Hardware-Umgebungen benötigen. Informationen zum Technischen Support finden Sie auf der SPSS Inc.-Website unter <http://www.spss.com> oder wenden Sie sich an Ihr regionales Büro, das Sie auf der SPSS Inc.-Website unter <http://www.spss.com/worldwide> finden. Beachten Sie, dass Sie nach Ihrem Namen, dem Namen Ihrer Organisation und Ihrer Seriennummer gefragt werden.

Kundenmeinungen

Ihre Meinung ist uns wichtig. Teilen Sie uns bitte Ihre Erfahrungen mit SPSS Inc.-Produkten mit. Senden Sie uns eine E-Mail an suggest@us.ibm.com oder schreiben Sie an: SPSS Inc., Attn: Director of Product Planning, 233 South Wacker Drive, 11th Floor, Chicago, IL 60606-6412.

1 IBM SPSS Collaboration and Deployment Services - Essentials for Python **1**

Übersicht	1
Installation	1
Installation unter Windows	1
Installation unter UNIX	2
Installation unter IBM i	3

2 Befehlszeilen-Skripts **4**

Globale Schlüsselwörter	4
Repository-Verbindungen	5
Content-Repository-Skripts	5
Schlüsselwörter	6
Vorgänge	7
Prozessmanagement-Funktionen	25
Schlüsselwörter	25
Vorgänge	25

3 Das PESImpl-Modul **29**

Content-Repository-API	29
Methoden	30
Wrapper-Klassen	54
Prozessmanagement-API	57
Methoden	57
Wrapper-Klassen	61

Anhang

A Notices

66

Index

68

IBM SPSS Collaboration and Deployment Services - Essentials for Python

Übersicht

IBM® SPSS® Collaboration and Deployment Services stellt ein Skript-Framework mit einigen APIs bereit, mit denen erfahrene Benutzer und Administratoren eigenständige Routinen oder Batch-Jobs 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 Bezeichnungen 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 dem IBM® SPSS® Collaboration and Deployment Services Repository in Ihrem eigenen Python-Knoten.

Allgemeine Informationen zu Python, eine dynamische, objektorientierte Programmiersprache, finden Sie auf der [Python-Website \(http://www.python.org\)](http://www.python.org).

Installation

Das Skript-Framework kann auf Windows-, UNIX- und IBM i-Plattformen installiert werden. Die Skriptplattform ist unabhängig von der Plattform, die vom Repository verwendet und auf die von der Skriptfunktion zugegriffen wird. Ein Repository, das z. B. auf der Windows-Plattform ausgeführt wird, kann von Skriptfunktionen aufgerufen werden, die auf der UNIX-Plattform ausgeführt werden.

Installation unter Windows

1. Wenn Python bereits auf Ihrem System installiert ist, deinstallieren Sie es.
2. Legen Sie das Installationsmedium ein.

3. Öffnen Sie das Verzeichnis *PYTHON\Disk1\InstData\NoVM* auf Datenträger 2 und doppelklicken Sie auf *install.exe*. Befolgen Sie die Anweisungen am Bildschirm, um die Installation auszuführen. Führen Sie die Installation für den Standardspeicherort durch. Damit werden die erforderlichen Python-, ZSI- und PyXML-Technologien installiert.
4. Öffnen Sie das Verzeichnis *PYTHON* auf dem Installationsmedium und extrahieren Sie den Inhalt von *pes-scripting-1.0.zip* in ein temporäres Verzeichnis.
5. Fügen Sie den Speicherort des IBM® SPSS® Collaboration and Deployment Services - Essentials for Python-Verzeichnisses der Systemumgebungsvariablen **Path** Ihres PCs hinzu.
6. Ändern Sie an der Befehlseingabeaufforderung das aktuelle Verzeichnis zu dem Ordner, in dem Sie *pes-scripting-1.0.zip* extrahiert haben. Geben Sie den folgenden Befehl ein und drücken Sie die Eingabetaste.

```
python setup.py install
```

Installation unter UNIX

1. Wenn Python 2.4.3, ZSI 2.0 rc3 und PyXML 0.8.4 noch nicht auf Ihrem System installiert sind, installieren Sie diese nach dem Download von der jeweiligen Website, bevor Sie mit Schritt 2 fortfahren.
 - Python 2.4.3: <http://www.python.org/download/releases/2.4.3/>
 - ZSI 2.0 rc3: <http://sourceforge.net/projects/pywebsvcs>
 - PyXML 0.8.4: http://sourceforge.net/project/showfiles.php?group_id=6473
2. Legen Sie den Datenträger 2 ein.
3. Öffnen Sie das Verzeichnis *PYTHON* und extrahieren Sie den Inhalt von *pes-scripting-1.0.tar.gz* in ein temporäres Verzeichnis.
4. Bearbeiten Sie *setup.cfg* im temporären Verzeichnis. Ersetzen Sie `<PythonInstallDir>` durch Ihren IBM® SPSS® Collaboration and Deployment Services - Essentials for Python-Installationspfad. Wenn kein Wert angegeben ist, weist der Pfad standardmäßig auf die Python-Bibliothek, zum Beispiel */usr/lib/python2.4*.

```
[install]
install-base = <PythonInstallDir>
install-data = <PythonInstallDir>
install-purelib = <PythonInstallDir>
install-scripts = <PythonInstallDir>
install_headers = <PythonInstallDir>
```

5. Ändern Sie an der Befehlseingabeaufforderung das aktuelle Verzeichnis zu dem Ordner, in dem Sie *pes-scripting-1.0.tar.gz* extrahiert haben. Geben Sie den folgenden Befehl ein:

```
python setup.py install
```

Beachten Sie, dass beim Extrahieren von Tar-Paketen einige Tar-Dienstprogramme Meldungen wie die folgende ausgeben:

```
tar: A lone zero block at ####
```

Der Wert von ##### entspricht einem ganzzahligen Wert. Diese Meldungen dienen lediglich zur Information und weisen nicht auf einen Fehler bei der Extraktion hin.

Installation unter IBM i

1. Melden Sie sich bei Ihrem IBM i-System über ein Telnet-Terminal an.
2. Legen Sie den Datenträger 1 ein.
3. Starten Sie QShell mit dem folgenden Befehl:

```
QSH
```

4. Ändern Sie das Verzeichnis in */qopt/Server/IBMi/Python*.
5. Kopieren Sie den Inhalt des Verzeichnisses an einen temporären Speicherort.
6. Führen Sie das Installationskript mit dem folgenden Befehl aus:

```
./PyInst.scr
```

Python wird als */QOpenSys/usr/local/bin/python2.4* installiert und IBM® SPSS® Collaboration and Deployment Services - Essentials for Python wird in */QOpenSys/usr/local/lib/python2.4/site-packages* installiert.

Befehlszeilen-Skripts

Die Python-Datei *CADSTool.py* kann aus der Befehlszeile heraus verwendet werden, um im IBM® SPSS® Collaboration and Deployment Services Repository gespeicherte Ressourcen zu bearbeiten. Die allgemeine Syntax, die verwendet wird, um IBM® SPSS® Collaboration and Deployment Services-Skriptoperationen aus der Befehlszeile aufzurufen, ist:

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

Erläuterung:

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

Globale Schlüsselwörter

Tabelle 2-1 gibt die Schlüsselwörter an, die von allen IBM® SPSS® Collaboration and Deployment Services-Skriptfunktionen unterstützt werden. Die zweite Spalte enthält optionale, abgekürzte Versionen der Schlüsselwörter. Beachten Sie, dass bei Schlüsselwörtern zwischen Groß- und Kleinbuchstaben unterschieden wird.

Tabelle 2-1
Globale Schlüsselwörter

Schlüsselwort	Optionale Kurzversion	Verwendung
--user	-u	Der Benutzername für die Verbindung mit dem Repository-Server.
--password	-p	Das Passwort für die Verbindung mit dem Repository-Server.
--host	-q	Der Name des Hosts/Servers, auf dem das Repository installiert ist
--port	-o	Die Portnummer des Repository-Servers
--useDefault	-z	Gibt an, dass Benutzer, Passwort, Host und Port aus der Datei <i>Authorization.properties</i> gelesen werden müssen
--ssl		Gibt an, dass der Repository-Server das Secure Socket Layer-(SSL-)Protokoll zur Verschlüsselung der Kommunikation verwendet. Vor der Verwendung dieses Schlüsselworts muss der Repository-Server für SSL konfiguriert sein. Weitere Informationen finden Sie in der Administratordokumentation.
-h		Hilfe-Informationen zum Skriptmodul

Repository-Verbindungen

Am Ende jedes Befehls müssen Sie Benutzer-ID, Passwort, Host und Port für das IBM® SPSS® Collaboration and Deployment Services Repository angeben. Folgende Methoden können verwendet werden, um diese Verbindungsinformationen bereitzustellen:

- Die Verwendung von Schlüsselwörtern, wie beispielsweise der folgenden:

```
--user <user> --password <password> --host <Host> --port <Port>
```

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

```
# Autorisierungsinformationen
user=admin
password=spss
host=yourhost
port=80
```

Parameter, die durch die Befehlszeile übergeben werden, haben stets Vorrang. Wenn beispielsweise `--user` und `--password` über die Befehlszeile bereitgestellt werden und außerdem der Parameter `--useDefault` oder der Parameter `-z` bereitgestellt wird, werden Benutzer und Passwort aus der Befehlszeile verwendet und Host und Port aus der Datei *Authorization.properties* abgerufen. Wenn dagegen Benutzer, Passwort, Host und Port über die Befehlszeile bereitgestellt werden, jedoch außerdem der Parameter `--useDefault` bzw. `-z` verwendet wird, wird `--useDefault` ignoriert und nur die Informationen aus der Befehlszeile werden 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-Skripts

Content-Repository-Skripts 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 Bezeichnungen, Sicherheit und Metadaten

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

Schlüsselwörter

Tabelle 2-2 gibt die Schlüsselwörter an, die für alle Repository-Funktionen unterstützt werden. Die zweite Spalte enthält optionale, abgekürzte Versionen der Schlüsselwörter.

Wichtiger Hinweis: Bei Schlüsselwörtern wird zwischen Groß- und Kleinschreibung unterschieden.

Tabelle 2-2
Schlüsselwörter für Repository-APIs

Schlüsselwort	Optionale Kurzversion	Verwendung
--source	-s	Der Pfad der Quelldatei oder des Quellordners
--target	-t	Der Pfad des Zielordners
--version	-v	Die Version einer Datei
--principal	-r	Der Benutzer, dem die Berechtigung zugewiesen werden muss
--permission	-n	Der Berechtigungstyp (z. B. Lesen, Schreiben, Ändern, Löschen)
--label	-l	Das Label, das einer Version einer Datei zugeordnet wird
--criteria	-c	Die Suchkriterien für die Suche nach Metadaten-Attributen von Dateien oder Ordnern
--author	-a	Der Name des Autors einer Datei oder eines Ordners
--description	-d	Die Beschreibung für eine Datei oder einen Ordner
--title	-i	Der Titel für eine Datei oder einen Ordner
--expirationDate	-q	Das Ablaufdatum für eine Datei oder einen Ordner
--expirationStartDate		Das Ablauf-Anfangsdatum für eine Datei oder einen Ordner
--expirationEndDate		Das Ablauf-Enddatum für eine Datei oder einen Ordner
--keyword	-k	Das Schlüsselwort für eine Datei oder einen Ordner
--cascade	-x	Gibt an, dass Sicherheitseinstellungen für einen Ordner an Unterordner und Dateien weitergeleitet 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 "topic1;topic2"
--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>Übergebene Jobs</i> gesucht werden soll.
--propertyName		Der Name einer benutzerdefinierten Eigenschaft

Schlüsselwort	Optionale Kurzversion	Verwendung
--customProperty		Der Name/das Wertepaar einer benutzerdefinierten Eigenschaft, die aktualisiert werden soll
--propertyName		Der Name der benutzerdefinierten Eigenschaft, für die gültige Werte abgerufen werden sollen

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

Vorgänge

Im folgenden Abschnitt sind alle Repository-Skriptoperationen aufgeführt, die in IBM® SPSS® Collaboration and Deployment Services unterstützt werden.

Der Vorgang "advanceSearch"

Sucht auf Basis verschiedener Parameter nach Dateien und Ordnern im Repository. Beachten Sie, dass `expirationStartDate` und `expirationEndDate` zurzeit nicht funktionieren, wenn sie gemeinsam mit anderen Suchfeldern wie Titel, Autor usw. verwendet werden.

Syntax

```
python CADSTool.py advanceSearch --author <author>
  --title <title> --description <Beschreibung>
  --createdBy <createdBy> --modifiedBy <modifiedBy>
  --keyword <keyword> --label <Label>
--topic <Thema>
--expirationStartDate <expirationStartDate>
--expirationEndDate <expirationEndDate>
--submittedHierarchy -z
```

Erläuterung:

- `<author>` ist der Name des Autors.
- `<title>` ist der Titel der Datei/des Ordners.
- `<description>` ist die Beschreibung der Datei/des Ordners.
- `<createdBy>` ist der Name des Benutzers, der die Datei/den Ordner erstellt hat.
- `<modifiedBy>` ist der Name des Benutzers, der die Datei/den Ordner geändert hat.
- `<keyword>` ist das der Datei/dem Ordner zugeordnete Schlüsselwort.
- `<label>` ist das Label für das Versionskennzeichen.
- `<topic>` ist das der Datei/dem Ordner zugeordnete Thema.
- `<expirationStartDate>` ist das Ablauf-Anfangsdatum der Datei/des Ordners. Das Datumsformat lautet `YYYY-MM-DDThh:mm:ss.sTZD` (z. B. `1997-07-16T19:20:30.45+01:00`), wobei gilt:
YYYY = vierstellige Jahresangabe

MM = zweistellige Monatsangabe (01 ist Januar usw.)
 DD = 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)
 s = Stellen, die einen Dezimalbruchteil einer Sekunde angeben, wobei der gültige Bereich zwischen 0 und 999 liegt
 TZD = Zeitzonenbezeichner (Z oder +hh:mm oder -hh:mm)

- `<expirationEndDate>` ist das Ablauf-Enddatum der Datei/des Ordners. Das Datumsformat lautet `YYYY-MM-DDThh:mm:ss.sTZD`.
- `--submittedHierarchy` gibt an, dass sich die Datei/der Ordner im Ordner "Übergebene Jobs" befindet.

Alle Parameter sind optional.

Beispiel

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

Nachrichten

Die folgenden Meldungen 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 für gewöhnlich die Dateinamen mit ihrem vollständig qualifizierten Pfad und ihren Versionen.
- Fehler bei der Suche nach Dateien und Ordnern

Der Vorgang "applySecurity"

Stellt die Sicherheits-Zugriffskontrollliste (ACL) für eine Datei bzw. einen Ordner im Repository ein.

Syntax

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

Erläuterung:

- `<source>` ist der vollständig qualifizierte IBM® SPSS® Collaboration and Deployment Services Repository-Pfad der Datei oder des Ordners, 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 Sicherheits-Provider, der für den Abruf von Informationen über die Benutzer (Principals) verwendet werden soll. Dies ist ein optionaler Parameter.
- `--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
```

Nachrichten

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

- `<permission>` Berechtigung erfolgreich definiert für `<source>`.
- `<source>` Datei oder Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<permission>` Ungültiger Berechtigungstyp, bitte versuchen Sie es erneut.
- `<source>` Fehler bei der Definition der Sicherheits-ACL.

Der Vorgang "cascadeSecurity"

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- Sicherheits-ACL erfolgreich übertragen für `<source>`.

- `<source>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler bei der Übertragung der Sicherheits-ACL.

Der Vorgang "copyResource"

Kopiert 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 Kopieren umbenannt werden kann. Die Fälle, die am Anfang von [Der Vorgang "moveResource" auf S. 19](#) 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 Pfad im Repository, 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` erfolgreich kopiert.
- `<source>` Datei oder Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<target>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Kopieren der Datei bzw. des Ordners.

Der Vorgang "createFolder"

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Ordner erfolgreich erstellt.
- `<source>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<folder>` Ordner existiert bereits. Bitte versuchen Sie es erneut.
- `<source>` Fehler bei der Erstellung des Ordners.

Der Vorgang "deleteFile"

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 "Übergebene 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 "Übergebene Jobs" gelöscht:

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

Nachrichten

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

- `<source>` erfolgreich gelöscht.

- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Löschen der Datei.

Der Vorgang "deleteFileVersion"

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 Bezeichnung der Datei, die gelöscht werden soll. Dies ist ein optionaler Parameter.
- `--submittedHierarchy` löscht eine spezifische Version einer Datei aus dem Ordner "Übergebene 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 Bezeichnung gelöscht:

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

Im folgenden Beispiel wird eine Datei mit einer spezifischen Bezeichnung aus dem Ordner "Übergebene 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` erfolgreich gelöscht.
- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Löschen der Datei.

Der Vorgang "deleteFolder"

deleteFolder 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 "Übergebene 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 "Übergebene Jobs" gelöscht:

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

Nachrichten

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

- <source> erfolgreich gelöscht.
- <source> Ordner existiert nicht. Bitte versuchen Sie es erneut.
- <source> Fehler beim Löschen des Ordners.

Der Vorgang "downloadFile"

Lädt eine 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 oder Objekt-URI des Ordners, der die Datei enthält, die heruntergeladen werden soll. Der Objekt-URI ist in den Eigenschaften eines Ordners in IBM® SPSS® Collaboration and Deployment Services Deployment Manager angegeben. Dies ist ein erforderlicher Parameter.
- <version> ist die Version der Datei, die heruntergeladen werden soll. Dies ist ein optionaler Parameter.

- `<label>` ist die Bezeichnung der Datei, die heruntergeladen werden soll. Dies ist ein optionaler Parameter.
- `<source>` 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 eine spezifische Version der Datei mithilfe eines Versionskennzeichens 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Datei erfolgreich heruntergeladen.
- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<target>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Herunterladen der Datei.

Der Vorgang "export"

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` erfolgreich exportiert.
- `<source>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Export des Ordners.

Der Vorgang "getAccessControlList"

Ruft die Sicherheits-Zugriffskontrollliste (ACL) 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Datei oder Ordner existiert nicht. Bitte versuchen Sie es erneut.
- Fehler beim Abrufen der Sicherheitsdetails für `<source>`.

Der Vorgang "getAllVersions"

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 "Übergebene 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 "Übergebene Jobs" abgerufen:

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

Nachrichten

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

- <source> Datei existiert nicht. Bitte versuchen Sie es erneut.
- <source> Fehler beim Abrufen der Dateiversionen.
- Wenn der Prozess erfolgreich abgeschlossen wird, werden die Informationen für jede Dateiversion angezeigt, darunter Versionskennzeichen und Label-Informationen.

Der Vorgang "getChildren"

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 der des Ordners. Dies ist ein erforderlicher Parameter.

Beispiel

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

Nachrichten

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

- Wenn der Befehl erfolgreich ausgeführt wird, wird der gesamte Inhalt des angegebenen Ordners angezeigt.
- <source> Ordner existiert nicht. Bitte versuchen Sie es erneut.
- <source> Fehler beim Abrufen der Ressourcen.

Der Vorgang "getCustomPropertyValue"

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- *<propertyName>* nimmt Werte an als *<valid values>*
- Fehler beim Abrufen der Eigenschaftsdetails für *<source>*.

Der Vorgang "getMetadata"

Ruft die Metadaten-Attribute 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-/Bezeichnungsattribute ignoriert. Dies ist ein erforderlicher Parameter.
- *<version>* ist die Version der Datei oder des Ordners, für die/den Metadaten abgerufen werden sollen. Dies ist ein optionaler Parameter.
- *<label>* ist die Bezeichnung der Datei oder des Ordners, für die/den Metadaten abgerufen werden sollen. Dies ist ein optionaler Parameter.
- `--submittedHierarchy` ruft Metadaten aus dem Ordner "Übergebene 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 "Übergebene Jobs" abgerufen:

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

Nachrichten

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

- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Abrufen der Dateimetadaten.
- Wenn der Prozess erfolgreich abgeschlossen wird, werden alle Metadateninformationen für die angegebene Datei oder den angegebenen Ordner angezeigt, darunter benutzerdefinierte Metadateneigenschaften.

Der Vorgang "import"

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

Syntax

```
python CADSTool.py import --source "<source>" --target "<target>" -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.

Beispiel

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

Nachrichten

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

- `<source>` erfolgreich importiert.
- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<target>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Import des Ordners.

Der Vorgang "moveResource"

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 Folder/Temp.txt* und das Ziel ist */Demo Folder* :

- **Fall 1:** Falls der Ordner *Demo Folder* existiert, wird *Temp.txt* in *Demo Folder* verschoben.
- **Fall 2:** Falls der Ordner *Demo Folder* nicht existiert, wird *Temp.txt* in " / " verschoben und in *Demo Folder* umbenannt.

Falls die Quelle */Temp Folder/Temp.txt* und das Ziel */Demo Folder/Abc.dat* ist:

- **Fall 1:** Falls der Ordner *Demo Folder* existiert, wird *Temp.txt* in *Demo Folder* verschoben und in *Abc.dat* umbenannt.
- **Fall 2:** Falls der Ordner *Demo Folder* nicht existiert, wird eine Fehlermeldung 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- *<source>* erfolgreich verschoben.
- *<source>* Datei oder Ordner existiert nicht. Bitte versuchen Sie es erneut.

- `<target>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Verschieben der Datei bzw. des Ordners.

Der Vorgang "removeLabel"

Entfernt eine Bezeichnung 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 Bezeichnung 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- Bezeichnung erfolgreich entfernt für `<source>`.
- `<source>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Löschen der Bezeichnung.
- `<label>` Bezeichnung existiert nicht. Bitte versuchen Sie es erneut.

Der Vorgang "removeSecurity"

Entfernt die Sicherheits-Zugriffskontrollliste (ACL) von 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 Sicherheits-Provider, der für den Abruf von Informationen über die Benutzer (Principals) verwendet werden soll. Dies ist ein optionaler Parameter.
- `--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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Gesamte Sicherheits-ACL erfolgreich entfernt.
- `<source>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Löschen der Sicherheits-ACL.

Der Vorgang "search"

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 Meldungen 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 für gewöhnlich die Dateinamen mit ihrem vollständig qualifizierten Pfad und ihren Versionen.
- `<criteria>` Keine Datei bzw. kein Ordner entspricht den Suchkriterien.
- Fehler bei der Suche nach Dateien und Ordnern

Der Vorgang "setLabel"

Fügt eine Bezeichnung zu einer Version einer Datei im Repository hinzu. Falls die Datei bereits beschriftet ist, wird die ursprüngliche Bezeichnung entfernt und durch die neue Bezeichnung 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 Bezeichnung hinzugefügt werden soll. Dies ist ein erforderlicher Parameter.
- <label> ist der Name der Bezeichnung, 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- Bezeichnung erfolgreich definiert für <source>.
- <source> Ordner existiert nicht. Bitte versuchen Sie es erneut.
- <source> Fehler bei der Definition der Bezeichnung.

Der Vorgang "setMetadata"

Fügt Metadateneigenschaften zu Dateien und Ordnern im Repository hinzu. [Tabelle 2-3](#) listet die Metadateneigenschaften auf und gibt an, ob sie Dateien und/oder Ordnern hinzugefügt werden können.

Tabelle 2-3

Metadateneigenschaften und Ressourcentypen

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 "<Beschreibung>"
--expirationDate "<expirationDate>" --topic "<Thema>" --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 YYYY-MM-DDThh:mm:ss.sTZD (z. B. 1997-07-16T19:20:30.45+01:00), wobei gilt:
 YYYY = vierstellige Jahresangabe
 MM = zweistellige Monatsangabe (01 ist Januar usw.)
 DD = 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)
 s = Stellen, die einen Dezimalbruchteil einer Sekunde angeben, wobei der gültige Bereich zwischen 0 und 999 liegt
 TZD = Zeitzonebezeichner (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 oder des Ordners, der/dem Metadaten hinzugefügt werden sollen. Dies ist ein optionaler Parameter.
- *<label>* ist die beschriftete Version der Datei oder des Ordners, der/dem 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 mehr als eine benutzerdefinierte Eigenschaft anzuwenden, verwenden Sie ein Semikolon (;) als Trennzeichen (<customProperty>=<value>;<customProperty>=<value>). Trennen Sie Mehrfachauswahl-Eigenschaftswerte mit dem Operator | (<customProperty>=opt1|opt2;<customProperty>=value).

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

Beispiel

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

```
--customProperty "multi=hi|hello|bye;Complexity Degree=Simple" -z
```

Nachrichten

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

- `<source>` Metadaten erfolgreich definiert.
- `<source>` Datei oder Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler bei der Definition der Metadaten.

Der Vorgang "uploadFile"

`uploadFile` speichert eine Datei aus dem lokalen Dateisystem im Content-Repository, wobei die Option besteht, eine neue Version der Datei zu erstellen, falls sie 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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Datei erfolgreich hochgeladen.
- `<source>` Datei existiert nicht. Bitte versuchen Sie es erneut.
- `<target>` Ordner existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Hochladen der Datei.

Prozessmanagement-Funktionen

Prozessmanagement-Skripts 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 Skripts für Prozessmanagement-Funktionen über die Python-Befehlszeile beschrieben. Jede API enthält detaillierte Syntaxinformationen, ein Beispiel und erwartete Meldungen.

Schlüsselwörter

[Tabelle 2-4](#) gibt die Schlüsselwörter an, die für alle Prozessmanagement-APIs unterstützt werden. Die zweite Spalte enthält optionale, abgekürzte Versionen der bereitgestellten Schlüsselwörter. Die Tabelle enthält nur Schlüsselwörter, die für Prozessmanagement-APIs spezifisch sind. Zusätzliche Schlüsselwörter, die sowohl für Prozessmanagement-APIs als auch für Repository-APIs gelten, finden Sie unter [Tabelle 2-1](#) und [Tabelle 2-2](#).

Tabelle 2-4
Schlüsselwörter für Prozessmanagement-APIs

Schlüsselwort	Optionale Kurzversion	Verwendung
--source	-s	Der Quell-Job 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.

Vorgänge

Im folgenden Abschnitt sind alle Prozessmanagement-Skript-APIs aufgeführt, die in IBM® SPSS® Collaboration and Deployment Services unterstützt werden. Die Syntax und die Beispiele, die unten angegeben sind, enthalten die Mindestanzahl an erforderlichen Parametern.

Der Vorgang "deleteJobExecutions"

Löscht die angegebenen Job-Ausführungs-Objekte.

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- Ausführungs-ID nicht angegeben.

Der Vorgang "executeJob"

Führt einen Job abhängig von den übergebenen Parametern synchron oder asynchron aus. Im Fall einer synchronen Ausführung kommt die API nicht wieder, bis der Job abgeschlossen ist. Im Fall einer asynchronen Ausführung kommt die API nach Beginn des Jobs wieder.

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 Meldungen können bei der Verwendung dieser API angezeigt werden:

- `<source>` Job erfolgreich ausgeführt. Die Job-Ausführungs-ID ist `<execId>`.
- `<source>` Job existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler bei der Ausführung des Jobs.

Der Vorgang "getJobExecutionDetails"

Zeigt Ausführungsdetails für einen spezifischen Job an, 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 Job-Protokoll inline angezeigt werden soll. Wenn der `--log`-Parameter 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 `--log`-Parameter verwendet.

Beispiele

Im folgenden Beispiel werden die Details einer spezifischen Job-Ausführung aufgelistet:

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

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

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

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

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

Nachrichten

Die folgenden Meldungen 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.
- `<source>` Ausführung existiert nicht. Bitte versuchen Sie es erneut.
- `<execId>` Fehler beim Anzeigen der Details einer Job-Ausführung.
- `--target` kann nicht ohne den `--log`-Parameter verwendet werden

Der Vorgang "getJobExecutionList"

Gibt die momentanen Ausführungen und die abgeschlossenen Ausführungen für einen spezifischen Job und für alle Versionen des Jobs an.

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 Meldungen 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>` Job existiert nicht. Bitte versuchen Sie es erneut.
- `<source>` Fehler beim Anzeigen der Ausführungsliste für einen Job.

Das *PESImpl*-Modul

Die IBM® SPSS® Collaboration and Deployment Services - Essentials for Pythonfunktion ermöglicht die Interaktion mit IBM® SPSS® Collaboration and Deployment Services Repository-Objekten direkt innerhalb von Python-Skripts. Importieren Sie innerhalb Ihres Python-Codes die *PESImpl*-Klasse aus dem `pes.api.PESImpl`-Modul. Erstellen Sie ein *PESImpl*-Objekt unter Verwendung der Verbindungsinformationen für das Repository, mit dem eine Verbindung hergestellt werden soll.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl(user, password, Host, Port, ssl=True)
```

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

- *user* entspricht dem Benutzernamen
- *password* entspricht dem Passwort, 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 Secure Socket Layer-(SSL-)Protokoll zur Verschlüsselung der Kommunikation verwendet. Wenn der Parameter “ssl” auf *False* gesetzt ist oder wenn der Parameter bei der Erstellung des *PESImpl*-Objekts weggelassen wird, wird für die Serverkommunikation nicht SSL verwendet. Bei Verwendung von *ssl=True* muss der Repository-Server für SSL konfiguriert sein. Weitere Informationen finden Sie in der Administratordokumentation.

Der Zugriff auf spezifische Methoden erfolgt über das `pesImpl`-Objekt.

Content-Repository-API

Content-Repository-Skripts 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 Bezeichnungen, 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 Meldungen.

Methoden

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

Anmerkungen:

- Verwenden Sie für alle Methoden mit den optionalen Parametern `Label` und `Version` entweder `Label` oder `Version`, nicht jedoch beide. Falls keine `Version` oder kein `Label` für eine bestimmte Datei bzw. einen bestimmten Ordner angegeben wird, wird die letzte `Version` verwendet.
- Für alle Methoden, die einen Pfad zu Dateien/Ordern im Repository benötigen, kann entweder der Pfad oder der Objekt-URI verwendet werden. Der Objekt-URI ist in den Objekteigenschaften in IBM® SPSS® Collaboration and Deployment Services Deployment Manager aufgeführt.
- Für Methoden, die die Eingabe von Pfaden für das Quell- oder Ziel-Repository oder das Dateisystem erfordern, welche nichtlateinische 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')
```

Die Methode "advanceSearch"

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

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
- Ablauf-Anfangsdatum
- Ablauf-Enddatum
- MIME-Typ
- Beschriftung
- Schlüsselwort
- Themen

Beachten Sie, dass `expirationStartDate` und `expirationEndDate` zurzeit nicht funktionieren, wenn sie gemeinsam mit anderen Suchfeldern wie Titel, Autor usw. verwendet werden.

Tabelle 3-1
Eingangsparameter für "advanceSearch"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>criteriaDict</i>	Erforderlich	Wörterbuch	Das Wörterbuch enthält den Schlüssel/Wert des Paares, anhand dessen die Suche ausgeführt wird. Die akzeptablen Schlüsselwerte sind: <ul style="list-style-type: none"> • author • title • Beschreibung • createdBy • modifiedBy • expirationStartDate • expirationEndDate • mimeType • Label • keyword • Thema 	{ "author": "admin", "title": "search", "label": "label 1", }
<i>submittedHierarchy</i>	Optional	Boolesch	Gibt an, ob im Ordner <i>Übergebene Jobs</i> gesucht werden soll.	True oder False.

Tabelle 3-2
Rückgabewert für "advanceSearch"

Typ	Beschreibung
PageResult	Struktur, in der jede Zeile mit einer Suchentsprechung übereinstimmt. Für weitere Informationen siehe Thema Die Klasse "PageResult" auf S. 56.

Tabelle 3-3
Ausnahmen für "advanceSearch"

Typ	Beschreibung
InsufficientParameterException	Obligatorische Parameter sind nicht angegeben.

Beispiel

Beim folgenden Beispielcode werden alle Dateiversionen mit dem Label *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()
```

Die Methode "applySecurity"

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

Stellt die Sicherheits-Zugriffskontrollliste (ACL) für eine Datei bzw. einen Ordner im Repository ein.

Tabelle 3-4
Eingangsparameter für "advanceSecurity"

Feld	Verwenden	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 Sicherheits-Provider, der verwendet wird, um Sicherheitseinstellungen auf Benutzer anzuwenden (z. B. <i>Native</i>)	Native
<i>cascade</i>	Optional	Boolesch	Überträgt die Sicherheitseinstellungen auf alle Dateien und Unterordner im jeweiligen Ordner	True oder False.

Tabelle 3-5
Rückgabewert für "applySecurity"

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

Tabelle 3-6
Ausnahmen für "applySecurity"

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

Beispiel

Im folgenden Beispiel wird dem Benutzer *icrod* die Berechtigung *READ* (Lesen) für die angegebene Datei zugewiesen.

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

Die Methode "cascadeSecurity"

```
cascadeSecurity(source)
```

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

Tabelle 3-7
Eingangsparameter für "cascadeSecurity"

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

Tabelle 3-8
Rückgabewert für "cascadeSecurity"

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

Tabelle 3-9
Ausnahmen für "cascadeSecurity"

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

Beispiel

Der folgende Beispielcode leitet die Sicherheit für den Ordner *Projects* an alle untergeordneten Objekte des Ordners weiter.

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

Die Methode "copyResource"

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

Kopiert eine Datei oder einen Ordner in einen anderen Ordner im Repository. Die angegebene Quelldatei bzw. der angegebene Quellordner kann beim Kopieren umbenannt werden. Weitere Informationen zum Umbenennen finden Sie unter [Die Methode "moveResource" auf S. 46.](#)

Tabelle 3-10
Eingangsparameter für "copyResource"

Feld	Verwenden	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 umzubenennen.	"/New Folder" oder "/New Folder/abc.dat"

Tabelle 3-11
Rückgabewert für "copyResource"

Typ	Beschreibung
Zeichenfolge	URI der kopierten Datei bzw. des kopierten Ordners

Tabelle 3-12
Ausnahmen für "copyResource"

Typ	Beschreibung
ResourceNotFoundException	Der Quell- 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
```

Die Methode "createFolder"

```
createFolder(source)
```

Erstellt einen neuen Ordner an einem festgelegten Speicherort im Repository.

Tabelle 3-13
Eingangsparameter für "createFolder"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der Ordner, der im Repository erstellt werden soll	"/Temp Folder/Sample Folder"

Tabelle 3-14
Rückgabewert für "createFolder"

Typ	Beschreibung
Zeichenfolge	URI des erstellten Ordners

Tabelle 3-15
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 Meldung 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."
```

Die Methode "deleteFile"

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

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

Tabelle 3-16
Eingangsparameter für "deleteFile"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"
<i>submitted-Hierarchy</i>	Optional	Boolesch	Zeigt an, ob die Datei sich im Ordner <i>Übergebene Jobs</i> befindet.	True oder False.

Tabelle 3-17
Rückgabewert für "deleteFile"

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

Tabelle 3-18
Ausnahmen für "deleteFile"

Typ	Beschreibung
ResourceNotFoundException	Die Quelldatei 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."
```

Die Methode "deleteFileVersion"

```
deleteFileVersion(source, Version, Label, submittedHierarchy)
```

Löscht eine bestimmte Version einer Datei aus dem Repository.

Tabelle 3-19
Eingangsparameter für "deleteFileVersion"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	" /Temp Folder/Temp.txt " oder "0a58c3670016a78600 00010dcee0eaa28219 "
<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 "

Feld	Verwenden	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>Übergebene Jobs</i> befindet.	True oder False.

Tabelle 3-20
Rückgabewert für "deleteFileVersion"

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

Tabelle 3-21
Ausnahmen für "deleteFileVersion"

Typ	Beschreibung
ResourceNotFoundException	Der Quell- 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 das Label *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")
```

Die Methode "deleteFolder"

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

Löscht einen Ordner und seine Inhalte aus dem Repository.

Tabelle 3-22
Eingangsparameter für "deleteFolder"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners im Repository	"/Temp Folder" oder "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Optional	Boolesch	Zeigt an, ob der Ordner sich im Ordner <i>Übergebene Jobs</i> befindet.	True oder False.

Tabelle 3-23
Rückgabewert für "deleteFolder"

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

Tabelle 3-24
Ausnahmen für "deleteFolder"

Typ	Beschreibung
ResourceNotFoundException	Der angegebene Ordner ist nicht vorhanden.
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
IllegalParameterException	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 Meldung über eine Ausnahme an die Konsole gesendet.

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

Die Methode "downloadFile"

```
downloadFile(source, target, Version, Label)
```

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

Tabelle 3-25
Eingangsparameter für "downloadFile"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Repository-Pfad oder Objekt-URI der herunterzuladenden Datei	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad (im lokalen Dateisystem) des Ordners, in den die Datei heruntergeladen werden soll	"C:\Temp"
<i>Version</i>	Optional. Es kann entweder eine Version oder ein 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 eine Version oder ein Label angegeben werden.	Zeichenfolge	Die spezifische beschriftete Version der Datei, die heruntergeladen werden soll.	"Version 2"

Tabelle 3-26
Rückgabewert für "downloadFile"

Typ	Beschreibung
Resource	Container für Informationen zu einem Repository-ObjektFür weitere Informationen siehe Thema Die Ressourcenklasse auf S. 54..

Tabelle 3-27
Ausnahmen für "downloadFile"

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

Beispiel

Mit dem folgenden Beispielcode wird eine mit dem Label *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")
```

Die Methode "exportResource"

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

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

Tabelle 3-28
Eingangsparameter für "exportResource"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Repository-Pfad oder Objekt-URI des zu exportierenden Ordners	"/Temp Folder" oder "0a58c3670016a78 60000010dcee0eaa2 8219"
<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 3-29
Rückgabewert für "exportResource"

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

Tabelle 3-30
Ausnahmen für "exportResource"

Typ	Beschreibung
ResourceNotFoundException	Der Quell- 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")
```

Die Methode "getAccessControlList"

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

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

Tabelle 3-31
Eingangsparameter für "getAccessControlList"

Feld	Verwenden	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 "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Optional	Boolesch	Gibt an, ob die Datei bzw. der Ordner sich im Ordner <i>Übergebene Jobs</i> befindet.	True oder False.

Tabelle 3-32
Rückgabewert für "getAccessControlList"

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

Tabelle 3-33
Ausnahmen für "getAccessControlList"

Typ	Beschreibung
ResourceNotFoundException	Der Quell- 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
```

Die Methode "getAllVersions"

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

Ruft eine Liste aller Versionen einer Datei im Repository ab.

Tabelle 3-34
Eingangsparameter für "getAllVersions"

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

Tabelle 3-35
Rückgabewert für "getAllVersions"

Typ	Beschreibung
Liste	Eine Liste von Ressourcenobjekten. Informationen dazu finden Sie unter Die Ressourcenklasse auf S. 54.

Tabelle 3-36
Ausnahmen für "getAllVersions"

Typ	Beschreibung
ResourceNotFoundException	Die Quelldatei ist nicht vorhanden.
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 Versionslabel 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()
```

Die Methode "getChildren"

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

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

Tabelle 3-37
Eingangsparameter für "getChildren"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI des Ordners im Repository	"/Temp Folder" oder "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Optional	Boole'scher Wert	Zeigt an, ob der Ordner sich im Ordner "Übergebene Jobs" befindet	True oder False.

Tabelle 3-38
Rückgabewert für "getChildren"

Typ	Beschreibung
Liste	Eine Liste von Ressourcenobjekten. Informationen dazu finden Sie unter Die Ressourcenklasse auf S. 54.

Tabelle 3-39
Ausnahmen für "getChildren"

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

Beispiel

Mit folgendem Beispielcode werden die untergeordneten Elemente des Ordners */Demo/Drafts* abgerufen und jeweils Titel, Autor und Ressourcenbezeichner 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()
```

Die Methode "getCustomPropertyValue"

`getCustomPropertyValue(propertyName)`

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

Tabelle 3-40
Eingangsparameter für "getCustomPropertyValue"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>propertyName</i>	Erforderlich	Zeichenfolge	Der Name der benutzerdefinierten Eigenschaft.	"FreeForm"

Tabelle 3-41
Rückgabewert für "getCustomPropertyValue"

Typ	Beschreibung
Liste	Gibt eine Liste gültiger Werte zurück, die von der benutzerdefinierten 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 3-42
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 *Sprache* zugegriffen.

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

Die Methode "getMetadata"

```
getMetadata(source, Version, Label, submittedHierarchy)
```

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

Tabelle 3-43
Eingangsparameter "getMetadata"

Feld	Verwenden	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>Version</i>	Optional. Es kann entweder eine Version oder ein Label angegeben werden.	Zeichenfolge	Die spezifische Version der Datei oder des Ordners	"0:2006-08-25 21:15:49.453"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>Label</i>	Optional. Es kann entweder eine Version oder ein 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>Übergebene Jobs</i> befindet.	True oder False.

Tabelle 3-44
Rückgabewert für "getMetadata"

Typ	Beschreibung
Resource	Container für Informationen zu einem Repository-Objekt. Für weitere Informationen siehe Thema Die Ressourcenklasse auf S. 54.

Tabelle 3-45
Ausnahmen für "getMetadata"

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

Beispiel

Im folgenden Beispiel wird auf den Ressourcenbezeichner 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()
```

Die Methode "importResource"

```
importResource(source, target)
```

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

Tabelle 3-46
Eingangsparameter für "importResource"

Feld	Verwenden	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 "0a58c3670016a7860000 010dcee0eaa28219"

Tabelle 3-47
Rückgabewert für "importResource"

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

Tabelle 3-48
Ausnahmen für "importResource"

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

Beispiel

Dier folgende Beispielcode.

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

Die Methode "moveResource"

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

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 3-49
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.	Fehlermeldung wird ausgegeben.

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

- Falls der Ordner *Demo Folder* existiert, wird *Temp.txt* in *Demo Folder* verschoben.
- Falls der Ordner *Demo Folder* nicht existiert, wird *Temp.txt* in “/“ verschoben und in *Demo Folder* umbenannt.

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

- Falls der Ordner *Demo Folder* existiert, wird *Temp.txt* in *Demo Folder* verschoben und in *Abc.dat* umbenannt.
- Falls der Ordner *Demo Folder* nicht existiert, wird eine Fehlermeldung angezeigt.

Tabelle 3-50
Eingangsparameter für “*moveResource*”

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichen- folge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei bzw. des Ordners im Repository	" /Temp Folder/Temp.txt " oder " 0a58c3670016a78600 00010dcee0eaa28219 "
<i>target</i>	Erforderlich	Zeichen- folge	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.	" /New Folder " oder " /New Folder/abc.dat "

Tabelle 3-51
Rückgabewert für “*moveResource*”

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

Tabelle 3-52
Ausnahmen für “*moveResource*”

Typ	Beschreibung
<code>ResourceNotFoundException</code>	Die angegebene Quelle ist nicht vorhanden.
<code>InsufficientParameterException</code>	Erforderliche Parameter sind nicht angegeben.

Beispiel

Dier folgende Beispielcode.

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

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

Die Methode "removeLabel"

```
removeLabel(source, Label)
```

Entfernt eine Bezeichnung von einer Datei im Repository.

Tabelle 3-53

Eingangsparameter für "removeLabel"

Feld	Verwenden	Typ	Beispielwert	Beschreibung
<i>source</i>	Erforderlich	Zeichenfolge	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000010dcee0eaa28219"	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository
<i>Label</i>	Erforderlich	Zeichenfolge	"Version 1"	Der Bezeichnungsname, der entfernt werden soll

Tabelle 3-54

Rückgabewert für "removeLabel"

Typ	Beschreibung
Zeichenfolge	URI der aktualisierten Datei

Tabelle 3-55

Ausnahmen für "removeLabel"

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

Beispiel

Der folgende Beispielcode entfernt das Label *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")
```

Die Methode "removeSecurity"

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

Entfernt die Sicherheits-Zugriffskontrollliste (ACL) von einer angegebenen Datei bzw. einem Ordner im Repository.

Tabelle 3-56
Eingangsparameter für "removeSecurity"

Feld	Verwenden	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
<i>provider</i>	Optional	Zeichenfolge	Der Sicherheits-Provider (z. B. <i>Native</i>), der verwendet wird, um Informationen über Benutzer zu erhalten.	Native
<i>cascade</i>	Optional	Boolesch	Überträgt die Sicherheitseinstellungen auf alle Dateien und Unterordner im jeweiligen Ordner	True oder False.

Tabelle 3-57
Rückgabewert für "removeSecurity"

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

Tabelle 3-58
Ausnahmen für "removeSecurity"

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

Beispiel

Der folgende Beispielcode entfernt die ACL für den Principal *icrod* aus der Datei *MyReport.rptdesign*.

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

Die Methode "search"

```
search(criteria)
```

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

Tabelle 3-59
Eingangsparameter für "search"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>criteria</i>	Erforderlich	Zeichenfolge	Der Wert, der für die Suche in den Datei-Metadaten verwendet wird	"Age"

Tabelle 3-60
Rückgabewert für "search"

Typ	Beschreibung
PageResult	Struktur, in der jede Zeile mit einer Suchentsprechung übereinstimmt. Für weitere Informationen siehe Thema Die Klasse "PageResult" auf S. 56.

Tabelle 3-61
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()
```

Die Methode "setLabel"

```
setLabel(source, Version, Label)
```

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

Tabelle 3-62
Eingangsparameter für "setLabel"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad oder Objekt-URI der Datei im Repository	"/Temp Folder/Temp.txt" oder "0a58c3670016a7860000 010dcee0eaa28219"
<i>Version</i>	Erforderlich	Zeichenfolge	Die spezifische Version der Datei	"0:2006-08-25 21:15:49.453"
<i>Label</i>	Erforderlich	Zeichenfolge	Die Bezeichnung, die zu der Datei hinzugefügt wird	"Version 1"

Tabelle 3-63
Rückgabewert für "setLabel"

Typ	Beschreibung
Zeichenfolge	URI der aktualisierten Datei

Tabelle 3-64
Ausnahmen für "setLabel"

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

Beispiel

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

Die Methode "setMetadata"

```
setMetadata(source, Version, Label, props)
```

Fügt Metadateneigenschaften zu Dateien und Ordnern im Repository hinzu. In der folgenden Datei werden die Metadateneigenschaften aufgeführt und angegeben, ob sie auf Dateien und/oder Ordner angewendet werden können.

Tabelle 3-65
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

Tabelle 3-66
Eingangsparameter für "setMetadata"

Feld	Verwenden	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>Version</i>	Optional. Es kann entweder eine Version oder ein 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 eine Version oder ein Label angegeben werden.	Zeichenfolge	Die Bezeichnung 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, nimmt sie die Liste als einen Wert aus <code>topic</code> und das Wörterbuch als <code>customProperty</code> . Für den Rest der Metadaten nimmt sie "string".	{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }

Tabelle 3-67
Rückgabewert für "setMetadata"

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

Tabelle 3-68
Ausnahmen für "setMetadata"

Typ	Beschreibung
InsufficientParameterException	Erforderliche Parameter sind nicht angegeben.
ResourceNotFoundException	Quelldatei- 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
```

Die Methode "uploadFile"

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

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

Tabelle 3-69
Eingangsparameter für "uploadFile"

Feld	Verwenden	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 3-70
Rückgabewert für "uploadFile"

Typ	Beschreibung
Zeichenfolge	URI der hochgeladenen Datei

Tabelle 3-71
Ausnahmen für "uploadFile"

Typ	Beschreibung
ResourceNotFoundException	Der Quell- oder Zielordner ist nicht vorhanden.

Typ	Beschreibung
ResourceAlreadyExistsException	Eine Datei bzw. ein Ordner mit dem Namen der Quelldatei ist im Zielordner vorhanden und der Parameter <code>versionFlag</code> 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 mithilfe 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
```

Wrapper-Klassen

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

Die Ressourcenklasse

Die Klasse `Resource` fungiert als vereinfachter Wrapper für das Repository-Objekt `ResourceSpecifier.Resource` und bietet Zugriff auf objektspezifische Informationen. Neben den Standard-Metadaten, die zu Repository-Objekten gehören, enthält diese Klasse auch etwaige benutzerdefinierte Metadaten-Informationen, die für Objekte im Repository definiert wurden.

[Tabelle 3-72](#) gibt alle Methoden an, die in der `Resource`-Klasse verfügbar sind.

Tabelle 3-72
Methoden für die Klasse "Resource"

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> die Berechtigung <i>Löschen</i> für <i>Ressource X</i> hat, gibt <code>getAccessControlList</code> des Ressourcenobjekts <code>X{ 'Joe': 'DELETE' }</code> zurück, nicht aber alle drei Berechtigungen aus dem Aufruf des Webdienstes.
<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.

Methodenname	Beschreibung
getContentSize	Gibt die Größe des Objekts zurück.
getCreatedBy	Gibt den Namen des Benutzers, der das Objekt erstellt hat, als Zeichenfolge zurück.
getCreationDate	Gibt das Erstellungsdatum des Objekts als Datetime-Objekt zurück.
getDescription	Gibt die Beschreibung des Objekts als Liste zurück.
getDescriptionLanguage	Gibt die Sprache des Objekts als Liste zurück.
getExpirationDate	Gibt das Ablaufdatum des Objekts als Datetime-Objekt zurück.
isExpired	Gibt an, ob das angegebene Objekt abgelaufen ist oder nicht
getMIMEType	Gibt den MIME-Typ des Objekts als Zeichenfolge zurück.
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 den Ressourcenbezeichner 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 das Label 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.

Die Klasse "IdentificationSpecifier"

Diese Klasse fungiert als vereinfachter Wrapper für das Repository-Objekt `IdentificationSpecifier` und bietet Zugriff auf informationsspezifische Daten für das Objekt. [Tabelle 3-73](#) gibt alle Methoden an, die in der `IdentificationSpecifier`-Klasse verfügbar sind.

Tabelle 3-73

Methoden für die Klasse "IdentificationSpecifier"

Methodenname	Beschreibung
getIdentifier	Gibt den ID-Wert eines Objekts als Zeichenfolge zurück.
getVersionMarker	Gibt die Version eines Objekts als Zeichenfolge zurück.
getVersionLabel	Gibt das zu einer Objektversion hinzugefügte Label als Zeichenfolge zurück.

Die 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 `PageResult`-Objekt mit vier Zeilen. [Tabelle 3-74](#) gibt alle Methoden an, die in der `PageResult`-Klasse verfügbar sind.

Tabelle 3-74
Methoden für die Klasse "PageResult"

Methodenname	Beschreibung
<code>getRows</code>	Gibt eine Liste von <code>SearchRow</code> -Objekten zurück. Für weitere Informationen siehe Thema Die Klasse "SearchRow" auf S. 56.

Die 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. [Tabelle 3-75](#) gibt alle Methoden an, die in der Klasse `SearchResult` verfügbar sind.

Tabelle 3-75
Methoden für die Klasse "SearchRow"

Methodenname	Beschreibung
<code>getTitle</code>	Gibt den Namen der Datei bzw. des Ordners zurück.
<code>getAuthor</code>	Gibt den Autor der Datei bzw. des Ordners zurück.
<code>getMimeType</code>	Gibt den MIME-Typ der Datei zurück.
<code>getObjectLastModifiedBy</code>	Gibt den Benutzer zurück, der die Datei bzw. den Ordner zuletzt geändert hat.
<code>getModified</code>	Gibt das Datum und die Uhrzeit der letzten Änderung der Datei bzw. des Ordners zurück.
<code>getFolderPath</code>	Gibt den Speicherort der Datei bzw. des Ordners zurück.
<code>getFolder</code>	Gibt den Namen des übergeordneten Ordners der Datei bzw. des Ordners zurück.
<code>getParentURI</code>	Gibt den Objekt-URI des übergeordneten Ordners zurück.
<code>getTopic</code>	Gibt die Themen zurück, die mit der Datei bzw. dem Ordner verknüpft sind.
<code>getChildRow</code>	Gibt die Liste mit <code>SearchChildRow</code> -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.

Die 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. [Tabelle 3-76](#) gibt alle Methoden an, die in der `SearchChildRow`-Klasse verfügbar sind.

Tabelle 3-76
Methoden für die Klasse "SearchChildRow"

Methodenname	Beschreibung
<code>getExpirationDate</code>	Gibt das Ablaufdatum der Datei bzw. des Ordners zurück.
<code>getKeyword</code>	Gibt die Schlüsselwörter zurück, die mit der Version der Datei bzw. des Ordners verknüpft sind.
<code>getVersionLabel</code>	Gibt das Versionslabel der Datei bzw. des Ordners zurück.
<code>getDescription</code>	Gibt die Beschreibung der Datei bzw. des Ordners zurück.
<code>getLanguage</code>	Gibt die Sprache zurück.
<code>getVersionCreationDate</code>	Gibt das Datum und die Uhrzeit der Erstellung der Datei bzw. des Ordners zurück.
<code>getVersionMarker</code>	Gibt die Versionsmarkierung der Datei bzw. des Ordners zurück.
<code>getUri</code>	Gibt den Objekt-URI der Datei bzw. des Ordners zurück.

Prozessmanagement-API

Prozessmanagement-Skripts 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 Meldungen.

Methoden

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

Hinweis: Für alle Methoden, die einen Pfad zu Dateien bzw. Ordnern im Repository benötigen, kann entweder der Pfad oder der Objekt-URI verwendet werden. Der Objekt-URI ist in den Objekteigenschaften in IBM® SPSS® Collaboration and Deployment Services Deployment Manager aufgeführt.

Die Methode "cancelJob"

```
cancelJob(executionId)
```

Bricht einen laufenden Job ab.

Tabelle 3-77

Eingangsparameter für "cancelJob"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Zeichenfolge	Ausführungs-ID für den Job	0a58c33d002ce908000010e0ccf7b01800e

Tabelle 3-78

Rückgabewert für "cancelJob"

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

Beispiel

In diesem Beispiel wird die Ausführung des *Berichte*-Jobs 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
```

Die Methode "deleteJobExecutions"

```
deleteJobExecutions(executionId)
```

Löscht eine oder mehrere Job-Ausführungen.

Tabelle 3-79

Eingangsparameter für "deleteJobExecutions"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Liste	Liste der Job-Ausführungs-IDs	[0a58c33d002ce90800000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

Tabelle 3-80

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 *Berichte* 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)

```

Die Methode "executeJob"

```
executeJob(source, Benachrichtigung, asynchronous)
```

Führt einen Job abhängig von den übergebenen Parametern synchron oder asynchron aus. Im Fall einer synchronen Ausführung kommt die Methode nicht wieder, bis der Job abgeschlossen ist. Im Fall einer asynchronen Ausführung kommt die Methode nach Beginn des Jobs wieder.

Tabelle 3-81

Eingangsparameter für "executeJob"

Feld	Verwenden	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>Benachrichtigung</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 3-82

Rückgabewert für "executeJob"

Typ	Beschreibung
Zeichenfolge	Das eindeutige Kennzeichen für die Ausführung des Jobs. Dieses Kennzeichen wird verwendet, um auf eine bestimmte Jobausführung zu verweisen.

Beispiel

In diesem Beispiel wird die Ausführung des *Berichte*-Jobs 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

```

Die Methode "getJobExecutionDetails"

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

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

Tabelle 3-83

Eingangsparameter für "getJobExecutionDetails"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>executionId</i>	Erforderlich	Zeichenfolge	Die Ausführungs-ID des Jobs	0a58c33d002ce9080000 010e0ccf7b01800e
<i>log</i>	Optional	Boolesch	Gibt an, ob das Job-Protokoll 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 3-84

Rückgabewert für "getJobExecutionDetails"

Typ	Beschreibung
jobExecutionDetails	Details zu einer Jobausführung. Für weitere Informationen siehe Thema Die Klasse "jobExecutionDetails" auf S. 62.

Beispiel

In diesem Beispiel werden Informationen zu Ausführungen von Jobschritten für die Jobausführung mit dem Kennzeichen *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()
```

Die Methode "getJobExecutionList"

```
getJobExecutionList(source)
```

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

Tabelle 3-85
Eingangsparameter für "JobExecutionList-API"

Feld	Verwenden	Typ	Beschreibung	Beispielwert
<i>source</i>	Erforderlich	Zeichenfolge	Der vollständig qualifizierte Pfad des Jobs im Repository	"/testJob"

Tabelle 3-86
Rückgabewert für "getJobExecutionList"

Typ	Beschreibung
PageResult	Container für die Liste mit Ausführungen für einen Job. Informationen dazu finden Sie unter Für weitere Informationen siehe Thema Die Klasse "PageResult" auf S. 61..

Beispiel

In diesem Beispiel werden die Ausführungen für den *Berichtejob* 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()
```

Wrapper-Klassen

Die PESImpl-API beinhaltet Klassen, die als Wrapper für Objekte fungieren, die von den durch die Prozessmanagement-Methoden aufgerufenen Webdiensten zurückgegeben werden. Diese Wrapper-Klassen bieten eine Oberfläche zur Anzeige der von den Methoden zurückgegebenen Informationen.

Die Klasse "PageResult"

Diese PageResult-Klasse 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 PageResult-Objekt mit vier Zeilen. [Tabelle 3-87](#) gibt alle Methoden an, die in der PageResult-Klasse verfügbar sind.

Tabelle 3-87
Methoden für die Klasse "PageResult"

Methodenname	Beschreibung
getRows	Gibt eine Liste mit Zeilenobjekten zurück, von denen eine jede die Ausführung eines Jobs darstellt. Für weitere Informationen siehe Thema Die Klasse "Row" auf S. 62.

Die Klasse "Row"

Die Row-Klasse dient als Container für Informationen zu einer Jobausführung auf Jobebene. Sie können mithilfe der Methoden dieser Klasse auf Metadaten zu einer Jobausführung zugreifen. [Tabelle 3-88](#) gibt alle Methoden an, die in der Row-Klasse verfügbar sind.

Tabelle 3-88
Methoden für die Klasse "Row"

Methodenname	Beschreibung
getObjId	Gibt die Ausführungs-ID des Jobs zurück.
getPath	Gibt den Pfad des Jobs zurück.
getVersionMarker	Gibt das Versionskennzeichen des Jobs zurück, der ausgeführt wurde.
getVersionLabel	Gibt die Versionsbezeichnung 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.

Die 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. [Tabelle 3-89](#) gibt alle Methoden an, die in der `jobExecutionDetails`-Klasse verfügbar sind.

Tabelle 3-89
 Methoden für die Klasse "jobExecutionDetails"

Methodenname	Beschreibung
getJobStepDetails	Gibt eine Liste von jobStepExecutionDetails-Objekten zurück. Für weitere Informationen siehe Thema Die Klasse "jobStepExecutionDetails" auf S. 63.
getArtifactLocation	Gibt eine Liste von Speicherorten von Job-Artefakten 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.
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.

Die Klasse "jobStepExecutionDetails"

Diese Klasse speichert die Ausführungsdetails für einen Jobschritt sowie eine Liste von jobStepChildExecutionDetails-Objekten. Diese Klasse enthält das ExecutionDetails-Objekt, an das sie all ihre Methodenaufrufe delegiert. [Tabelle 3-90](#) gibt alle Methoden an, die in der jobStepExecutionDetails-Klasse verfügbar sind.

Tabelle 3-90
 Methoden für die Klasse "jobStepExecutionDetails"

Methodenname	Beschreibung
getJobStepChldExecutionList	Gibt eine Liste von jobStepChildExecutionDetails-Objekten zurück. Für weitere Informationen siehe Thema Die Klasse "jobStepChildExecutionDetails" auf S. 64.
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/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 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.
getUserName	Gibt den Namen des Benutzers zurück, der den Jobschritt ausgeführt hat.
getUUID	Gibt die Ausführungs-ID des Jobschritts zurück.

Die Klasse "jobStepChildExecutionDetails"

Die jobStepChildExecutionDetails-Klasse 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. [Tabelle 3-91](#) gibt alle Methoden an, die in der jobStepChildExecutionDetails-Klasse verfügbar sind.

Tabelle 3-91
 Methoden für die Klasse "jobStepChildExecutionDetails"

Methodenname	Beschreibung
getArtifactLocation	Gibt eine Liste von untergeordneten Ausführungen von Job-Artefakten 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.

Notices

Licensed Materials – Property of SPSS Inc., an IBM Company. © Copyright SPSS Inc. 2004, 2010..

Patent No. 7,023,453

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: SPSS INC., AN IBM COMPANY, PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. SPSS Inc. may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-SPSS and non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this SPSS Inc. product and use of those Web sites is at your own risk.

When you send information to IBM or SPSS, you grant IBM and SPSS a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Information concerning non-SPSS products was obtained from the suppliers of those products, their published announcements or other publicly available sources. SPSS has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-SPSS products. Questions on the capabilities of non-SPSS products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to SPSS Inc., for the purposes of developing,

using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. SPSS Inc., therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. SPSS Inc. shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

SPSS is a trademark SPSS and Showcase are trademarks of SPSS Inc., an IBM Company, registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM, SPSS, or other companies.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.



ACL, 32, 40, 48
advanceSearch-Methode, 30
advanceSearch-Vorgang, 7
applySecurity-Methode, 32
applySecurity-Vorgang, 8

Beschriftungen
entfernen, 48
zuweisen, 50

cancelJob-Method, 58
cascadeSecurity-Methode, 33
cascadeSecurity-Vorgang, 9
copyResource-Methode, 33
copyResource-Vorgang, 10
createFolder-Methode, 34
createFolder-Vorgang, 10

Dateien
exportieren, 40
Herunterladen, 38
Hochladen, 53
Importieren, 45
kopieren, 33
löschen, 35
Metadaten, 44
Verschieben, 46
Versionen, 41

Dateiversionen
löschen, 36
deleteFile-Methode, 35
deleteFile-Vorgang, 11
deleteFileVersion-Methode, 36
deleteFileVersion-Vorgang, 12
deleteFolder-Methode, 37
deleteFolder-Vorgang, 13
deleteJobExecutions-Methode, 58
deleteJobExecutions-Vorgang, 25
downloadFile-Methode, 38
downloadFile-Vorgang, 13

executeJob-Methode, 59
executeJob-Vorgang, 26
export-Vorgang, 14
exportResource-Methode, 40

getAccessControlList-Methode, 40
getAccessControlList-Vorgang, 15
getAllVersions-Methode, 41
getAllVersions-Vorgang, 15

getChildren-Method, 42
getChildren-Vorgang, 16
getCustomPropertyValue-Methode, 43
getCustomPropertyValue-Vorgang, 16
getJobExecutionDetails-Methode, 60
getJobExecutionDetails-Vorgang, 27
getJobExecutionList-Methode, 60
getJobExecutionList-Vorgang, 28
getMetadata-Methode, 44
getMetadata-Vorgang, 17

IBM i, 3
IdentificationSpecifier-Klasse, 55
import-Vorgang, 18
importResource-Methode, 45
Installation, 1
IBM i, 3
UNIX, 2
Windows, 1

jobExecutionDetails-Klasse, 62
Jobs
Abbrechen, 58
Ausführen, 59
Ausführungen, 58, 60
Ausführungsdetails, 60
jobStepChildExecutionDetails-Klasse, 64
jobStepExecutionDetails-Klasse, 63

legal notices, 66

moveResource-Methode, 46
moveResource-Vorgang, 19

Ordner
erstellen, 34
kopieren, 33
löschen, 37
Metadaten, 44
Untergeordnete Elemente, 42
Verschieben, 46

PageResult-Klasse, 31, 50, 56, 61
PESImpl-Modul, 29

removeLabel-Methode, 48
removeLabel-Vorgang, 20
removeSecurity-Methode, 48
removeSecurity-Vorgang, 20

Ressourcenklasse, 39, 42–43, 45, 54
Row-Klasse, 62

search-Methode, 49
search-Vorgang, 21
SearchChildRow-Klasse, 57
SearchRow-Klasse, 56
Secure Sockets Layer, 29
setLabel-Methode, 50
setLabel-Vorgang, 22
setMetadata-Methode, 51
setMetadata-Vorgang, 22
SSL, 29
Suchen
 erweitert, 30

trademarks, 67

UNIX, 2
uploadFile-Methode, 53
uploadFile-Vorgang, 24

Windows, 1

Zugriffskontrolllisten
 Abrufen, 40
 entfernen, 48
 zuweisen, 32