

IBM SPSS Collaboration  
and Deployment Services –  
Essentials for Python 4.2 リ  
ファレンス



Note: Before using this information and the product it supports, read the general information under Notices p. 73 .

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..**

---

# はじめに

IBM® SPSS® Collaboration and Deployment Services は企業レベルのアプリケーションであり、予測分析の幅広い使用および展開を可能にします。IBM SPSS Collaboration and Deployment Services によって、分析資産を安全かつ監査可能な形で集中的に格納でき、予測分析プロセスの管理および制御を行う高度な機能が提供されます。また、分析処理の結果をエンドユーザーに提供する高度なメカニズムも提供されます。IBM SPSS Collaboration and Deployment Services を使用すると、分析資産の価値を保護し、法令順守を実現、分析の生産性を向上させ管理分析にかかる IT コストを最小化します。

## テクニカル サポート

SPSS Inc. のユーザー登録を行ったお客様は、SPSS Inc. のテクニカルサポートをご利用いただけます。SPSS Inc. 製品の使用方法、または対応するハードウェア環境へのインストールについてサポートが必要な場合は、テクニカルサポートにご連絡ください。テクニカルサポートに連絡するには、SPSS Inc. ホームページ (<http://www.spss.co.jp>) をご覧になるか、SPSS Inc. 社までお問い合わせください。お客様の ID、所属する組織 ID、およびシステムのシリアル番号をお手元にご用意ください。

## ご意見をお寄せください

お客様のご意見は貴重な情報です。SPSS Inc. 製品に関するご意見、ご感想をお寄せください。E-mail: [jpsales@spss.com](mailto:jpsales@spss.com) 郵便: 〒150-0012 東京都渋谷区広尾 1-1-39 恵比寿プライムスクエアタワー 10F エス・ピー・エス・エス株式会社。

---

# 内容

<b>1</b>	<b>IBM SPSS Collaboration and Deployment Services – Essentials for Python</b>	<b>1</b>
	概要 . . . . .	1
	インストール . . . . .	1
	Windows へのインストール . . . . .	2
	UNIX へのインストール . . . . .	2
	IBM i へのインストール . . . . .	3
<b>2</b>	<b>コマンドラインによるスクリプト</b>	<b>4</b>
	グローバル キーワード . . . . .	4
	リポジトリ接続 . . . . .	5
	コンテンツ リポジトリ スクリプト . . . . .	6
	キーワード . . . . .	6
	操作 . . . . .	7
	プロセス管理機能 . . . . .	27
	キーワード . . . . .	27
	操作 . . . . .	28
<b>3</b>	<b>PESImpl モジュール</b>	<b>32</b>
	コンテンツ リポジトリ API . . . . .	32
	メソッド . . . . .	33
	Wrapper クラス . . . . .	60
	プロセス管理 API . . . . .	63
	メソッド . . . . .	64
	Wrapper クラス . . . . .	68

## 付録

A Notices 73

索引 76



# IBM SPSS Collaboration and Deployment Services – Essentials for Python

## 概要

IBM® SPSS® Collaboration and Deployment Services には、PI のセットを使用できるスクリプト フレームワークが用意されています。経験のあるユーザーや管理者は、このフレームワークを使用して単独のルーチンを記述したり、一連のルーチンを組み合わせてリポジトリ オブジェクトやジョブを処理するバッチ ジョブを記述できます。次のような一括タスクを大幅に単純化できます。

- ユーザーの大きなグループに対するセキュリティ権限の変更
- 多くのフォルダまたはファイルのラベルづけまたはラベルの削除
- 多くのフォルダまたはファイルのアップロードまたはダウンロード

フレームワークには、コマンド ラインからタスクを実行する機能や、Python コード内で IBM® SPSS® Collaboration and Deployment Services Repository を操作できる高機能な API も含まれています。

動的なオブジェクト指向のプログラミング言語である Python の一般情報については、[Python サイト \(http://www.python.org\)](http://www.python.org) を参照してください。

## インストール

スクリプト フレームワークは、Windows、Unix、および IBM i の各プラットフォーム上にインストールできます。スクリプト プラットフォームは、スクリプト機能がアクセスするリポジトリで使用されるプラットフォームから独立しています。たとえば、Windows プラットフォームで実行するリポジトリは、UNIX プラットフォームで実行するスクリプト機能によって呼び出すことができます。

## Windows へのインストール

1. Python がシステムにインストールされている場合は、アンインストールします。
2. インストール メディアを挿入します。
3. Disk 2 の PYTHON¥Disk1¥InstData¥NoVM ディレクトリを開き、install.exe をダブルクリックします。画面の指示に従い、インストールを完了します。デフォルトの場所にインストールします。これにより、必要な Python、ZSI、および PyXML 技術がインストールされます。
4. インストール メディアの PYTHON ディレクトリを開き、pes-scripting-1.0.zip の内容を一時ディレクトリに展開します。
5. IBM® SPSS® Collaboration and Deployment Services - Essentials for Python ディレクトリの場所を PC の **Path** システム環境変数に追加します。
6. コマンドプロンプトで、現在のディレクトリを pes-scripting-1.0.zip を展開したフォルダに変更します。次のコマンドを入力して、Enter キーを押します。

```
python setup.py install
```

## UNIX へのインストール

1. Python 2.4.3、ZSI 2.0 rc3、および PyXML 0.8.4 がシステムにインストールされていない場合は、ステップ 2 に進む前に、それぞれの Web サイトからダウンロードした後、インストールします。
  - 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](http://sourceforge.net/project/showfiles.php?group_id=6473)
2. Disk 2 を挿入します。
3. PYTHON ディレクトリを開き、pes-scripting-1.0.tar.gz の内容を一時ディレクトリに展開します。
4. 一時ディレクトリで、setup.cfg を編集します。<PythonInstallDir> を IBM® SPSS® Collaboration and Deployment Services - Essentials for Python のインストールパスに変更します。指定されている値がない場合、パスは /usr/lib/python2.4 などの Python ライブラリにデフォルト指定されます。

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

5. コマンドプロンプトで、現在のディレクトリを pes-scripting-1.0.tar.gz を展開したフォルダに変更します。次のコマンドを実行します。

```
python setup.py install
```

tar バンドルを展開する場合、いくつかの tar コーティリティに次のようなメッセージが表示される場合があります。

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

##### の値は、整数値に対応します。これらのメッセージは情報を表示するだけのもので、展開の失敗を示すものではありません。

## IBM i へのインストール

1. Telnet ターミナルを使用して IBM i システムにログインします。
2. Disk 1 を挿入します。
3. 次のコマンドで QShell を開始します。

```
QSH
```

4. ディレクトリを /qopt/Server/IBMi/Python に変更します。
5. ディレクトリの内容を一時的な場所にコピーします。
6. 次のコマンドを実行して、インストール スクリプトを開始します。

```
./PyInst.scr
```

Python が /QOpenSys/usr/local/bin/python2.4 としてインストールされ、IBM® SPSS® Collaboration and Deployment Services – Essentials for Python が /QOpenSys/usr/local/lib/python2.4/site-packages にインストールされます。

# コマンド ラインによるスクリプト

Python ファイル CADSTool.py をコマンドラインから使用して、IBM® SPSS® Collaboration and Deployment Services Repository 内に保存されたリソースを操作できます。コマンドラインから IBM® SPSS® Collaboration and Deployment Services スクリプト操作を呼び出すために使用する一般的なシンタックスには、次のものがあります。

```
python CADSTool.py<操作> <キーワード>
```

ここでは次の表記を使用します。

- <Operation> は、呼び出す関数を示します
- <Keywords> は、関数への入力パラメータとして使用されるキーワードと値のペアを定義します

## グローバル キーワード

テーブル 2-1 は、すべての IBM® SPSS® Collaboration and Deployment Services スクリプト機能で使用できるキーワードのリストです。2 番目の列は、任意で使用できる短縮形のキーワードです。キーワードは大文字と小文字が区別されることに注意してください。

テーブル 2-1  
グローバル キーワード

キーワード	短縮形	使用方法
--user	-u	リポジトリ サーバーに接続するユーザー名
--password	-p	リポジトリ サーバーに接続するためのパスワード
--host	-q	リポジトリがインストールされているホストまたはサーバーの名前
--port	-o	リポジトリ サーバーのポート番号
--useDefault	-z	Authorization.properties ファイルからユーザー、パスワード、ホスト、ポートを読み取る必要があることを指定します。

キーワード	短縮形	使用方法
--ssl		リポジトリ サーバーが通信の暗号化に SSL プロトコルを使用していることを示します。ssl=True を使用する前に、リポジトリ サーバーに SSL を設定する必要があります。詳細は、管理者マニュアルを参照してください。
-h		スクリプト モジュールのヘルプ情報

## リポジトリ接続

すべてのコマンドの最後に、IBM® SPSS® Collaboration and Deployment Services Repository ユーザー ID、パスワード、ホスト、ポートを指定する必要があります。次の方法で、この接続情報を提供できます。

- 次のようなキーワードを使用する。

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

- コマンドに --useDefault パラメータ（または短縮形の -z）が含まれる Authorization.properties ファイルから指定する。この場合、Authorization.properties ファイルから接続情報が取得されます。このファイルは、<Scripting folder>%Lib%site-packages%config%Authorization.properties にあります。テキスト エディタを使用して、ファイル内の次の値をリポジトリの設定と一致する値に変更します。

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

コマンドラインから渡されたパラメータは、常に優先されます。たとえば、--user および --password をコマンドラインから指定し、--useDefault または -z パラメータも指定した場合は、コマンドラインからのユーザーとパスワードが使用され、Authorization.properties ファイルから取得されたホストとポートが適用されます。または、ユーザー、パスワード、ホストおよびポートがすべてコマンドラインを使用して提供されているが --useDefault または「-z パラメータを使用されている場合、--useDefault は無視され、コマンドライン情報だけが使用されます。

ここで説明されているすべての API について、シンタックスと例では、最小限の数のパラメータにするため -z パラメータを使用しています。

## コンテンツ リポジトリ スクリプト

コンテンツ リポジトリ スクリプトには、ファイルやフォルダなどのリポジトリ リソースを処理する機能があります。この領域には次の機能があります。

- フォルダの作成および削除
- ファイルのアップロードおよびダウンロード
- フォルダのエクスポートとインポート
- ラベルの管理、セキュリティ、およびメタデータ

この項では、Python コマンド ラインのリポジトリ機能用のスクリプトの使用方法について説明します。すべての操作には、詳細なシンタックス情報、例、予測されるメッセージが含まれています。

## キーワード

テーブル 2-2 は、リポジトリ機能で使用できるキーワードのリストです。2 番目の列は、任意で使用できる短縮形のキーワードです。

重要： キーワードは大文字と小文字が区別されます。

テーブル 2-2  
リポジトリ API 用のキーワード

キーワード	短縮形	使用方法
--source	-s	ソース ファイルまたはフォルダのパス
--target	-t	ターゲット フォルダのパス
--version	-v	ファイルのバージョン
--principal	-r	許可を付与する必要がある対象のユーザー
--permission	-n	許可のタイプ (読み取り、書き込み、変更、削除など)
--label	-l	ファイルのバージョンに割り当てられるラベル
--criteria	-c	ファイルまたはフォルダのメタデータ属性を検索するための検索条件
--author	-a	ファイルまたはフォルダの著者の名前
--description	-d	ファイルまたはフォルダの説明
--title	-i	ファイルまたはフォルダのタイトル
--expirationDate	-q	ファイルまたはフォルダの有効期限
--expirationStartDate		ファイルまたはフォルダの有効期間の開始日
--expirationEndDate		ファイルまたはフォルダの有効期間の終了日
--keyword	-k	ファイルまたはフォルダのキーワード

キーワード	短縮形	使用方法
--cascade	-x	フォルダのセキュリティ設定をサブフォルダとファイルに反映することを指定します。
--provider	-f	プリンシパルを取得するために使用するセキュリティプロバイダ
--createVersion	-b	ファイルの新しいバージョンを作成することを指定します。
--contentLanguage	-g	ファイルまたはフォルダのコンテンツ言語
--topic		ファイルまたはフォルダに割り当てられたトピック。--topic "topic1;topic2" のように複数の値を入力できます。
--modifiedBy		ファイルまたはフォルダを変更したユーザー
--mimeType		ファイルの MIME タイプ
--createdBy		ファイルまたはフォルダを作成したユーザー
--submittedHierarchy		[送信済みジョブ] フォルダを検索するかどうかを指定します。
--propertyName		カスタム プロパティの名前
--customProperty		更新するカスタム プロパティの名前と値のペア
--propertyName		有効な値を取得するカスタム プロパティの名前

ラベルとバージョン情報を使用できるすべての操作について、ユーザーはラベルまたはバージョンのいずれかを指定する必要がありますが、両方は指定できません。特定のファイルにバージョンとラベルの両方が指定されていない場合、最新のバージョンが使用されます。

## 操作

次の項は、IBM® SPSS® Collaboration and Deployment Services で使用できるすべてのリポジトリ スクリプト操作のリストです。

### advanceSearch 操作

さまざまなパラメータに基づき、リポジトリ内にあるファイルやフォルダを検索します。現在、expirationStartDate および expirationEndDate は、他の検索フィールド（タイトルや著者など）と組み合わせて使用した場合は機能しません。

#### シンタックス

```
python CADSTool.py advanceSearch --author<author>
```

```

--title<title> --description<description>
--createdBy<createdBy > --modifiedBy<modifiedBy>
--keyword<keyword> --label<label>
--topic<topic>
--expirationStartDate<expirationStartDate>
--expirationEndDate<expirationEndDate>
--submittedHierarchy -z

```

ここでは次の表記を使用します。

- <author> は、著者の名前です。
- <title> はファイルまたはフォルダのタイトルです。
- <description> はファイルまたはフォルダの説明です。
- <createdBy> は、ファイルまたはフォルダを作成したユーザーの名前です。
- <modifiedBy> は、ファイルまたはフォルダを変更したユーザーの名前です。
- <keyword> は、ファイルまたはフォルダに関連付けられているキーワードです。
- <label> は、バージョン マーカーのラベルです。
- <topic> は、ファイルまたはフォルダに関連付けられているトピックです。
- <expirationStartDate> は、ファイルまたはフォルダの有効期限の開始日です。日付の形式は、YYYY-MM-DDThh:mm:ss.sTZD (例：1997-07-16T19:20:30.45+01:00) です。詳細は次のようになります。  
 YYYY = 4 桁の年  
 MM = 2 桁の月 (01 は 1 月、など)  
 DD = 2 桁の日 (01 ~ 31)  
 hh = 2 桁の時間 (00 ~ 23、午前/午後の指定はなし)  
 mm = 2 桁の分 (00 ~ 59)  
 ss = 2 桁の秒 (00 ~ 59)  
 s = 秒の小数を表す桁数で、有効な範囲は 0 から 999  
 TZD = タイム ゾーンの指定 (z または +hh:mm または -hh:mm)
- <expirationEndDate> は、ファイルまたはフォルダの有効期限の終了日です。日付の形式は、YYYY-MM-DDThh:mm:ss.sTZD です。
- --submittedHierarchy は、ファイルまたはフォルダが [送信済みジョブ] フォルダにあることを示します。

すべてのパラメータは省略可能です。

## 例

```

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

```

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- API が正常に完了した場合、検索条件に一致したすべてのファイルとフォルダのリストが表示されます。通常、リストにはファイル名、完全修飾パス、およびバージョンが含まれます。
- ファイルおよびフォルダの検索中にエラーが発生しました

## applySecurity 操作

セキュリティ アクセス コントロール リスト (ACL) をリポジトリ内のファイルまたはフォルダに設定します。

## シンタックス

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

ここでは次の表記を使用します。

- <source> は、セキュリティ ACL を適用するファイルまたはフォルダの完全修飾 IBM® SPSS® Collaboration and Deployment Services Repository パスです。このパラメータは必須です。
- <principal> は、ACL の一部として指定したファイルまたはフォルダに適用されるユーザー (たとえば、admin など) です。このパラメータは必須です。
- <permission> は、指定したファイルまたはフォルダに適用される許可の種類です (読み取り、書き込み、変更、削除など)。このパラメータは必須です。
- <provider> は、ユーザー (プリンシパル) についての情報を取得するために使用するセキュリティ プロバイダです。このパラメータは省略可能です。
- --cascade は、フォルダにセキュリティを設定するときに、指定したフォルダ内のすべてのファイルとサブフォルダにも設定を適用する場合に使用します。このパラメータは省略可能です。

## 例

次の例は、フォルダにセキュリティを適用します。

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

次の例は、フォルダと、このフォルダ内のすべてのファイルおよびサブフォルダにセキュリティを適用します。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<permission>` 許可が `<source>` に対して正常に設定されました。
- `<source>` 指定したファイルまたはフォルダが存在しません。再試行してください。
- `<permission>` 許可の種類が無効です。再試行してください。
- `<source>` セキュリティ ACL の設定中にエラーが発生しました。

## cascadeSecurity 操作

フォルダのセキュリティ設定を、このフォルダ内のすべてのファイルとサブフォルダに適用します。

### シンタックス

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

`<source>` の値は、リポジトリ内のフォルダの完全修飾パスです。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` にセキュリティ ACL が正常にカスケードされました。
- `<source>` 指定したフォルダが存在しません。再試行してください。
- `<source>` セキュリティ ACL のカスケード中にエラーが発生しました。

## copyResource 操作

ファイルまたはフォルダをリポジトリ内の他のフォルダにコピーします。この API には名前変更機能があり、指定したファイルまたはフォルダをコピーするときに名前を変更できます。「[moveResource 操作](#)」( p. 20 ) の最初に説明されている例も、この `copyResource` API に適用されます。

## シンタックス

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

ここでは次の表記を使用します。

- <source> は、コピーするファイルまたはフォルダのコンテンツ リポジトリの完全修飾パスです。このパラメータは必須です。
- <target> は、ファイルまたはフォルダのコピー先のリポジトリの完全修飾パスです。このパラメータは必須です。

## 例

次の例は、ファイルをコピーします。

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

次の例は、ファイルをコピーして名前を変更します。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常にコピーされました。
- <source> 指定したファイルまたはフォルダが存在しません。再試行してください。
- <target> 指定したフォルダが存在しません。再試行してください。
- <source> ファイルまたはフォルダをコピー中にエラーが発生しました。

## createFolder 操作

リポジトリ内の指定した場所に新しいフォルダを作成します。

## シンタックス

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

<source> は、作成する新しいフォルダの完全修飾パスです。このパラメータは必須です。指定されたパスに基づき、サブフォルダを含む新しいフォルダが作成されます。

## 例

次の例では、Drafts を作成します (このフォルダが存在しない場合)。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> フォルダが正常に作成されました。
- <source> 指定したフォルダが存在しません。再試行してください。
- <folder> フォルダが既に存在します。再試行してください。
- <source> フォルダを作成中にエラーが発生しました。

### deleteFile 操作

リポジトリから、すべてのバージョンを含むファイルを削除します。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、削除するファイルの完全修飾リポジトリパスです。このパラメータは必須です。
- --submittedHierarchy は、[送信済みジョブ] フォルダからファイルを削除します。このパラメータは省略可能です。

### 例

次の例では、リポジトリから、すべてのバージョンを含むファイルを削除します。

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

次の例では、[送信済みジョブ] フォルダから、すべてのバージョンを含むファイルを削除します。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常に削除されました。
- <source> 指定したファイルが存在しません。再試行してください。
- <source> ファイルの削除中にエラーが発生しました。

## deleteFileVersion 操作

リポジトリから、ファイルの特定のバージョンを削除します。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、削除するファイルの完全修飾リポジトリパスです。このパラメータは必須です。
- <version> は、削除するファイルの特定のバージョンです。このパラメータは省略可能です。
- <label> は、削除するファイルのラベルです。このパラメータは省略可能です。
- --submittedHierarchy は、[送信済みジョブ] フォルダから特定のバージョンのファイルを削除します。このパラメータは省略可能です。

### 例

次の例では、特定のバージョンのファイルを削除します。

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

次の例では、特定のラベルが付いたファイルを削除します。

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

次の例では、[送信済みジョブ] フォルダから、特定のラベルが付いたファイルを削除します。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常に削除されました。
- <source> 指定したファイルが存在しません。再試行してください。
- <source> ファイルの削除中にエラーが発生しました。

## deleteFolder 操作

deleteFolder は、リポジトリからフォルダとフォルダ内のすべてのコンテンツを削除します。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、削除するフォルダの完全修飾リポジトリパスです。このパラメータは必須です。
- --submittedHierarchy は、[送信済みジョブ] フォルダから特定のバージョンのフォルダを削除します。このパラメータは省略可能です。

### 例

次の例では、フォルダを削除します。

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

次の例では、[送信済みジョブ] フォルダからフォルダを削除します。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常に削除されました。
- <source> 指定したフォルダが存在しません。再試行してください。
- <source> フォルダを削除中にエラーが発生しました。

## downloadFile 操作

リポジトリから、特定のバージョンのファイルをローカルファイルシステムにダウンロードします。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、ダウンロードするファイルを含むフォルダの完全修飾リポジトリパスまたはオブジェクトURIです。オブジェクトURIは、IBM® SPSS® Collaboration and Deployment Services Deployment Managerでフォルダのプロパティを表示することにより取得できます。このパラメータは必須です。
- <version> は、ダウンロードするファイルのバージョンです。このパラメータは省略可能です。
- <label> は、ダウンロードするファイルのラベルです。このパラメータは省略可能です。
- <target> は、ローカルファイルシステム上のファイルのダウンロード先となる場所の完全修飾リポジトリパスです。

## 例

次の例では、ファイルの最新バージョンをダウンロードします。

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

次の例では、バージョン マーカーを使用して特定のバージョンのファイルをダウンロードします。

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

次の例では、ラベルが付いたバージョンのファイルをダウンロードします。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> ファイルが正常にダウンロードされました。
- <source> 指定したファイルが存在しません。再試行してください。
- <target> 指定したフォルダが存在しません。再試行してください。
- <source> ファイルのダウンロード中にエラーが発生しました。

## export 操作

コンテンツ リポジトリからエクスポートを開始します。これにより、エクスポートするファイルおよびフォルダを選択したり、ローカルファイルシステム上に \*.pes エクスポート ファイルを保存することができます。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、エクスポートするフォルダの完全修飾リポジトリパスです。このパラメータは必須です。
- <target> は、作成する \*.pes エクスポート ファイルのローカルファイルシステム上の完全修飾パスです。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常にエクスポートされました。
- <source> 指定したフォルダが存在しません。再試行してください。
- <source> フォルダをエクスポート中にエラーが発生しました。

## getAccessControlList 操作

コンテンツ リポジトリ内の指定したファイルまたはフォルダのセキュリティ アクセス コントロール リスト (ACL) を取得します。

### シンタックス

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

<source> の値は、ファイルまたはフォルダの完全修飾パスです。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> 指定したファイルまたはフォルダが存在しません。再試行してください。
- <source> のセキュリティの詳細を取得中にエラーが発生しました。

## getAllVersions 操作

リポジトリ内のファイルのすべてのバージョンのリストを取得します。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、バージョンを取得する対象ファイルの完全修飾リポジトリパスです。このパラメータは必須です。
- --submittedHierarchy は、[送信済みジョブ] フォルダからバージョンを取得します。このパラメータは省略可能です。

### 例

次の例では、指定したファイルのすべてのバージョンを取得します。

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

次の例では、[送信済みジョブ] フォルダから、指定したファイルのすべてのバージョンを取得します。

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> 指定したファイルが存在しません。再試行してください。
- <source> ファイルのバージョンを取得中にエラーが発生しました。
- プロセスが正常に終了すると、バージョン マーカーとラベル情報を含むすべてのファイルバージョンの情報が表示されます。

## getChildren 操作

リポジトリの指定したフォルダ内のすべてのファイルおよびフォルダのリストを取得します。

### シンタックス

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

<source> の値は、ファイルまたはフォルダの完全修飾パスです。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- コマンドが正常に完了すると、指定したフォルダのすべてのコンテンツが一覧表示されます。
- `<source>` 指定したフォルダが存在しません。再試行してください。
- `<source>` リソースを取得中にエラーが発生しました。

## getCustomPropertyValue 操作

指定したカスタム プロパティで使用できる有効な値を取得します。

### シンタックス

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

`<propertyName>` の値は、カスタム プロパティの名前です。このパラメータは省略可能です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<propertyName>` は `<valid values>` として値を受け取ります
- `<propertyName>` のプロパティの詳細を取得中にエラーが発生しました。

## getMetadata 操作

リポジトリ内のファイルまたはフォルダのメタデータ属性を取得します。

### シンタックス

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

ここでは次の表記を使用します。

- `<source>>` は、メタデータを取得するファイルまたはフォルダの完全修飾リポジトリパスです。フォルダの場合は、バージョンまたはラベル属性は無視されます。このパラメータは必須です。
- `<version>` は、メタデータを取得するファイルまたはフォルダのバージョンです。このパラメータは省略可能です。
- `<label>` は、メタデータを取得するファイルまたはフォルダのラベルです。このパラメータは省略可能です。
- `--submittedHierarchy` は、[送信済みジョブ] フォルダからメタデータを取得します。このパラメータは省略可能です。

## 例

次の例は、フォルダのメタデータを取得します。

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

次の例では、ラベル付きバージョンのファイルのメタデータを取得します。

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

次の例では、[送信済みジョブ] フォルダ内のラベル付きバージョンのファイルのメタデータを取得します。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` 指定したファイルが存在しません。再試行してください。
- `<source>` ファイルのメタデータを取得中にエラーが発生しました。
- プロセスが正常に終了すると、指定したファイルまたはフォルダのカスタムメタデータプロパティを含むすべてのメタデータ情報が表示されます。

## import 操作

既存の \*.pes エクスポート ファイルをローカル ファイル システムからリポジトリにインポートします。

## シンタックス

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

ここでは次の表記を使用します。

- <source> は、リポジトリにインポートする \*.pes エクスポート ファイルのローカル ファイル システム上の完全修飾パスです。このパラメータは必須です。
- <target> は、\*.pes エクスポート ファイルのインポート先の完全修飾リポジトリパスです。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> は正常にインポートされました。
- <source> 指定したファイルが存在しません。再試行してください。
- <target> 指定したフォルダが存在しません。再試行してください。
- <source> フォルダをインポート中にエラーが発生しました。

### moveResource 操作

ファイルまたはフォルダをリポジトリ内の他のフォルダに移動します。この API には名前変更機能があり、指定したファイルまたはフォルダを移動するときに名前を変更できます。次の例で、名前変更機能の動作を説明します。

ソースが /Temp Folder/Temp.txt で、ターゲットが /Demo Folder の場合、次のようになります。

- **ケース 1**：フォルダ Demo Folder が存在する場合、Temp.txt は Demo Folder に移動します。
- **ケース 2**：フォルダ Demo Folder が存在しない場合、Temp.txt は “ / “ に移動し、名前が Demo Folder に変更されます。

ソースが /Temp Folder/Temp.txt で、ターゲットが /Demo Folder/Abc.dat の場合、次のようになります。

- **ケース 1**：フォルダ Demo Folder が存在する場合、Temp.txt は Demo Folder に移動し、名前が Abc.dat に変更されます。
- **ケース 2**：フォルダ Demo Folder が存在しない場合、エラーが表示されます。

### シンタックス

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

ここでは次の表記を使用します。

- `<source>` は、移動するファイルまたはフォルダの完全修飾リポジトリパスです。このパラメータは必須です。
- `<target>` は、ファイルまたはフォルダの移動先の場所の完全修飾リポジトリパスです。このパラメータは必須です。

## 例

次の例では、ファイルを移動します。

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

次の例では、フォルダを移動します。

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

次の例では、ファイルを移動して名前を変更します。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` は正常に移動されました。
- `<source>` 指定したファイルまたはフォルダが存在しません。再試行してください。
- `<target>` 指定したフォルダが存在しません。再試行してください。
- `<source>` ファイルまたはフォルダを移動中にエラーが発生しました。

## removeLabel 操作

リポジトリ内のファイルからラベルを削除します。

### シンタックス

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

ここでは次の表記を使用します。

- `<source>` は、リポジトリ内のファイルの完全修飾パスです。このパラメータは必須です。
- `<label>` は、指定したファイルから削除するラベルの名前です。このパラメータは必須です。

## 例

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

```
--label "Draft" -z
```

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- ラベルが <source> から正常に削除されました。
- <source> 指定したフォルダが存在しません。再試行してください。
- <source> ラベルの削除中にエラーが発生しました。
- <label> 指定したラベルが存在しません。再試行してください。

## removeSecurity 操作

リポジトリ内の指定したファイルまたはフォルダからセキュリティ アクセス コントロール リスト (ACL) を削除します。

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、セキュリティを削除するファイルまたはフォルダの完全修飾パスです。このパラメータは必須です。
- <principal> は、指定したファイルまたはフォルダからセキュリティを削除するユーザーまたはプリンシパル (たとえば、admin など) です。このパラメータは必須です。
- <provider> は、ユーザー (プリンシパル) についての情報を取得するために使用するセキュリティ プロバイダです。このパラメータは省略可能です。
- --cascade は、フォルダからセキュリティを削除するときに、指定したフォルダ内のすべてのファイルおよびサブフォルダからセキュリティ設定を削除する場合に使用します。このパラメータは省略可能です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- <source> すべてのセキュリティ ACL が正常に削除されました。

- `<source>` 指定したフォルダが存在しません。再実行してください。
- `<source>` セキュリティ ACL を削除中にエラーが発生しました。

## search 操作

リポジトリ内のファイルとフォルダを検索します。結果は、検索条件に一致したファイルまたはフォルダのリストとそのバージョンです。

### シンタックス

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

`<criteria>` は、リポジトリ内のすべてのファイルとフォルダのメタデータの検索に使用する検索文字列です。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- 検索が正常に完了すると、検索条件に一致したすべてのファイルとフォルダのリストが表示されます。通常、リストにはファイル名、完全修飾パス、およびバージョンが含まれます。
- `<criteria>` 検索条件に一致するファイルまたはフォルダはありません。
- ファイルおよびフォルダの検索中にエラーが発生しました。

## setLabel 操作

リポジトリ内の任意のバージョンのファイルにラベルを適用します。ファイルに既にラベルが付いている場合は、元のラベルは削除され、新しいラベルに置き換えられます。

### シンタックス

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

ここでは次の表記を使用します。

- `<source>` は、リポジトリ内のファイルの完全修飾パスです。このパラメータは必須です。

- <version> は、ラベルを適用するファイルのバージョンです。このパラメータは必須です。
- <label> は、指定したバージョンのファイルに適用されるラベルの名前です。このパラメータは必須です。

### 例

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

### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- ラベルが <source> に正常に設定されました。
- <source> 指定したフォルダが存在しません。再試行してください。
- <source> ラベルの設定中にエラーが発生しました。

## setMetadata 操作

メタデータ プロパティをリポジトリのファイルとフォルダに適用します。[テーブル 2-3](#) はメタデータ プロパティとファイルおよびフォルダのどちら（または両方）に適用されるかを示しています。

テーブル 2-3  
メタデータのプロパティとリソースのタイプ

メタデータのプロパティ	リソースのタイプ
著者	ファイル
説明	ファイルまたはフォルダ
タイトル	ファイルまたはフォルダ
有効期限	ファイルまたはフォルダ
キーワード	ファイル
トピック	ファイル
カスタム メタデータ	ファイルまたはフォルダ

### シンタックス

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

ここでは次の表記を使用します。

- <source> は、メタデータを設定するファイルまたはフォルダの完全修飾リポジトリパスです。このパラメータは必須です。
- <author> は、ファイルまたはフォルダの著者です。このパラメータは省略可能です。
- <title> はファイルまたはフォルダのタイトルです。このパラメータは省略可能です。
- <description> はファイルまたはフォルダの説明です。このパラメータは省略可能です。
- <expirationDate> は、ファイルまたはフォルダの有効期限の日付です。このパラメータは省略可能です。日付の形式は、YYYY-MM-DDThh:mm:ss.sTZD (例: 1997-07-16T19:20:30.45+01:00) です。詳細は次のようになります。  
 YYYY = 4 桁の年  
 MM = 2 桁の月 (01 は 1 月、など)  
 DD = 2 桁の日 (01 ~ 31)  
 hh = 2 桁の時間 (00 ~ 23、午前/午後の指定はなし)  
 mm = 2 桁の分 (00 ~ 59)  
 ss = 2 桁の秒 (00 ~ 59)  
 s = 秒の小数を表す桁数で、有効な範囲は 0 から 999  
 TZD = タイムゾーンの指定 (z または +hh:mm または -hh:mm)
- <keyword> は、ファイルまたはフォルダのキーワードです。このパラメータは省略可能です。
- <version> は、メタデータを適用するファイルまたはフォルダの特定のバージョンです。このパラメータは省略可能です。
- <label> は、メタデータを適用するファイルまたはフォルダのラベルが付いたバージョンです。このパラメータは省略可能です。
- <topic> は、ファイルまたはフォルダに適用するトピックです。このパラメータは省略可能です。
- <customProperty> は、ファイルまたはフォルダに適用するカスタムプロパティ値です。このパラメータは省略可能です。値は、<customProperty>=<value> のように指定します。複数のカスタムプロパティを適用するには、セミコロン (;) をセパレータとして使用します (<customProperty>=<value>;<customProperty>=<value>)。複数選択のプロパティ値は、| 演算子で区切ります (<customProperty>=opt1|opt2;<customProperty>=value)。

注: setMetadata API を使用するには、オプションのパラメータを少なくとも 1 つ指定する必要があります。

## 例

```
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
```

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` メタデータが正常に設定されました。
- `<source>` 指定したファイルまたはフォルダが存在しません。再試行してください。
- `<source>` メタデータの設定中にエラーが発生しました。

## uploadFile 操作

uploadFile は、ファイルをローカル ファイル システムからコンテンツ リポジトリに保存します。ファイルが既に存在する場合は、オプションとして新しいバージョンのファイルを作成できます。

## シンタックス

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

ここでは次の表記を使用します。

- `<source>` は、アップロードするファイルのローカル ファイル システム上の完全修飾パスです。このパラメータは必須です。
- `<target>` は、ファイルのアップロード先のリポジトリ内にあるフォルダの完全修飾パスです。このパラメータは必須です。
- `--createVersion` は、指定したファイルが存在し、新しいバージョンを作成することを示します。このパラメータは省略可能です。

## 例

次の例で、target は Drafts の完全修飾パスです。

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

MyReport.rptdesign がすでに /Demo/Drafts フォルダにある場合、次のように `--createVersion` パラメータを使用します。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` ファイルが正常にアップロードされました。
- `<source>` 指定したファイルが存在しません。再試行してください。

- <target> 指定したフォルダが存在しません。再試行してください。
- <source> ファイルのアップロード中にエラーが発生しました。

## プロセス管理機能

プロセス管理スクリプトには、ジョブを処理する機能があります。この領域には次の機能があります。

- ジョブの実行
- ジョブ履歴の取得
- ジョブ詳細の取得

この項では、プロセス管理機能用のスクリプトの Python コマンドラインでの使用方法について説明します。すべての API には、詳細なシンタックス情報、例、予測されるメッセージが含まれています。

## キーワード

テーブル 2-4 は、プロセス管理 API で使用できるキーワードのリストです。2 番目の列は、任意で使用できるキーワードの短縮形です。この表では、プロセス管理 API に固有のキーワードのみを挙げています。プロセス管理 API とリポジトリ API の両方に適用される追加のキーワードについては、以下を参照してください。テーブル 2-1 そして テーブル 2-2。

テーブル 2-4  
プロセス管理 API 用キーワード

キーワード	短縮形	使用方法
--source	-s	パスを含むソース ジョブ
--target	-t	ターゲット フォルダのパス
--notification	-j	ジョブの実行で通知を送信することを指定します。
--async	-m	ジョブを非同期で実行することを指定します。
--execId	-Y	ジョブの実行 ID
--jobStepName	-q	ジョブ ステップの名前
--log		ログを削除しないことを指定します。 --target と組み合わせて指定した場合は、ログは --target で指定した場所に保存されます。指定しない場合、ログはインラインで表示されます。

## 操作

次の項では、IBM® SPSS® Collaboration and Deployment Services で使用できるすべてのプロセス管理スクリプト API のリストです。ここで示すシンタックスと例には、最小限必要なパラメータが含まれています。

### deleteJobExecutions 操作

指定されたジョブ実行オブジェクトを削除します。

#### シンタックス

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

<execIDs> の値は実行が削除される識別子のスペース区切りリストです。このパラメータは必須です。

#### 例

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

#### メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- 実行 ID が指定されていません。

### executeJob 操作

指定したパラメータに基づき、ジョブを同期または非同期で実行します。同期で実行する場合は、ジョブが完了するまで API は返されません。非同期で実行する場合は、ジョブが開始した後に API が返されます。

#### シンタックス

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

ここでは次の表記を使用します。

- <source> は、リポジトリ内のジョブの完全修飾パスです。このパラメータは必須です。
- --notification は、ジョブの実行を通知する場合に使用します。このパラメータは省略可能です。
- --async は、ジョブを非同期で実行する場合に使用します。このパラメータは省略可能です。

## 例

次の例では、ジョブの実行を通知せず、ジョブを同期で実行します。

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

次の例では、ジョブの実行を通知し、ジョブを同期で実行します。

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

次の例では、ジョブの実行を通知せず、ジョブを非同期で実行します。

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

次の例では、ジョブの実行を通知し、ジョブを非同期で実行します。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- `<source>` ジョブは正常に実行されました。ジョブ実行 ID は `<execId>` です。
- `<source>` 指定したフォルダが存在しません。再試行してください。
- `<source>` ジョブの実行中にエラーが発生しました。

## getJobExecutionDetails 操作

特定のジョブのすべてのジョブ ステップと繰り返しを含む実行の詳細を一覧表示します。

## シンタックス

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

ここでは次の表記を使用します。

- `<execId>` は、ジョブの実行 ID です。このパラメータは必須です。
- `--log` は、ジョブのログをインラインで表示することを指定します。`--log` パラメータが含まれていない場合は、ジョブ ステップの実行により生成されたログは表示されません。このパラメータは省略可能です。
- `<target>` は、ログを保存するローカル ファイル システム上の場所です。このパラメータは省略可能で、使用する場合は `--log` パラメータを指定する必要があります。

## 例

次の例では、特定ジョブの実行の詳細を一覧表示します。

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

次の例では、特定ジョブの実行の詳細を一覧表示し、ログをインラインで表示します。

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

次の例では、特定ジョブの実行の詳細を一覧表示し、ジョブのログは特定の場所に保存されます。

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- 正常に終了した実行については、ジョブ、ジョブ ステップ、ジョブの繰り返しのすべての実行の詳細が一覧表示されます。ログはインラインで表示されるか、ローカル ファイル システム上の特定の場所に保存されます。
- `<execId>` 指定した実行が存在しません。再試行してください。
- `<execId>` ジョブの実行の詳細の表示中でエラーが発生しました。
- `--target` は、`--log` パラメータなしで使用することはできません。

## getJobExecutionList 操作

ジョブのすべてのバージョンについて、特定のジョブの現在の実行と完了した実行を一覧表示します。

### シンタックス

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

`<source>` の値は、リポジトリ内のジョブの完全修飾パスです。このパラメータは必須です。

## 例

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

## メッセージ

この API を使用する場合、次のメッセージが表示されることがあります。

- 指定したジョブの実行が正常に終了した場合は、実行 ID、ジョブ名、ジョブ実行ステータス、ジョブ実行の開始時刻と終了時刻などのすべての実行の詳細が一覧表示されます。
- `<source>` 指定したフォルダが存在しません。再試行してください。
- `<source>` ジョブの実行リストの表示でエラーが発生しました。

# PESImpl モジュール

IBM® SPSS® Collaboration and Deployment Services - Essentials for Python 機能を使用すると、Python スクリプト内で直接 IBM® SPSS® Collaboration and Deployment Services Repository オブジェクトを操作できます。Python コード内で、`pes.api.PESImpl` モジュールから `PESImpl` クラスをインポートします。接続先リポジトリの接続情報を使用して `PESImpl` オブジェクトを作成します。

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

`PESImpl` コンストラクタのパラメータは、次のとおりです。

- `user` はユーザー名に対応します。
- `password` は指定したユーザーに関連するパスワードに対応します。
- `host` はリポジトリ サーバーの名前を指定します。
- `port` はリポジトリ サーバーのポート番号を指定します。
- `ssl=True` は、リポジトリ サーバーが通信の暗号化に SSL プロトコルを使用していることを示します。 `ssl` パラメータが `False` に設定されている場合、`PESImpl` オブジェクト作成時にパラメータが省略されている場合、サーバー通信で SSL は使用されません。 `ssl=True` を使用する場合、リポジトリ サーバーに SSL を設定する必要があります。詳細は、管理者マニュアルを参照してください。

`pesImpl` オブジェクトを使用して、特定のメソッドにアクセスできるようになります。

## コンテンツ リポジトリ API

コンテンツ リポジトリ スクリプトには、ファイルやフォルダなどのリポジトリ リソースを処理する機能があります。この領域には次の機能があります。

- フォルダの作成および削除
- ファイルのアップロードおよびダウンロード
- フォルダのエクスポートとインポート
- ラベルの管理、セキュリティ、およびメタデータ

この項では、リポジトリに保存されたリソースの操作に使用する PESImpl API について概要を説明します。すべてのメソッドには、詳細なシンタックス情報、例、予測されるメッセージが含まれています。

## メソッド

次の項は、IBM® SPSS® Collaboration and Deployment Services でサポートされているすべてのコンテンツ リポジトリ メソッドのリストです。

注:

- オプション パラメータ `Label` および `Version` を持つメソッドについては、`Label` または `Version` のいずれか一方を使用し、両方は使用しないでください。特定のフォルダまたはファイルに `Version` と `Label` のどちらも指定されていない場合、最新のバージョンが使用されます。
- リポジトリのファイルまたはフォルダへのパスを必要とするすべてのメソッドについては、パスまたはオブジェクト URI のいずれかを使用できます。オブジェクト URI は、IBM® SPSS® Collaboration and Deployment Services Deployment Manager でオブジェクト プロパティを表示することにより取得できます。
- ラテン文字以外の Unicode 文字を使用するソース リポジトリまたは対象リポジトリ、またはファイル システム パスの入力が必要なメソッドの場合、次のように文字列を Unicode オブジェクトとして指定する必要があります。

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

### advanceSearch メソッド

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

入力として渡されるさまざまなパラメータに基づき、リポジトリ内にあるファイルやフォルダを検索します。次のアイテムを検索できます。

- 著者
- 説明
- タイトル
- 作成者 :
- 更新者
- 有効期限開始日
- 有効期限終了日
- MIME タイプ

## 3 章

- ラベル
- キーワード
- トピック

現在、`expirationStartDate` および `expirationEndDate` は、他の検索フィールド（タイトルや著者など）と組み合わせて使用した場合は機能しません。

テーブル 3-1  
advanceSearch の入力パラメータ

フィールド	使用	種類	説明	値の例
criteriaDict	必須	Dictionary	この辞書には、検索対象となるキー/値のペアが格納されます。使用可能なキーは次のとおりです。 <ul style="list-style-type: none"> <li>・ author</li> <li>・ title</li> <li>・ description</li> <li>・ createdBy</li> <li>・ modifiedBy</li> <li>・ expirationStartDate</li> <li>・ expirationEndDate</li> <li>・ mimeType</li> <li>・ label</li> <li>・ keyword</li> <li>・ topic</li> </ul>	<pre>{   "author": "admin",   "title": "search",   "label": "label 1", }</pre>
submittedHierarchy	オプション	ブール	[送信済みジョブ] フォルダを検索するかどうかを指定します。	True または False

テーブル 3-2  
advanceSearch の戻り値

種類	説明
PageResult	行が検索の一致に対応する構造。詳細は、p. 62 <a href="#">PageResult クラス</a> を参照してください。

テーブル 3-3  
advanceSearch の例外

種類	説明
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、Quarterly のキーワード値がある Production というラベルの 5 つのすべてのバージョンを返します。

```
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()
```

## applySecurity メソッド

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

セキュリティ アクセス コントロール リスト (ACL) をリポジトリ内のファイルまたはフォルダに設定します。

テーブル 3-4  
applySecurity の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000010dcee0eaa28219"
principal	必須	文字列	ACL の一部として指定したファイルまたはフォルダに適用されるユーザー (admin など) です。	admin
permission	必須	文字列	指定したファイルまたはフォルダに適用されるアクセス許可のタイプです。	READ、WRITE、DELETE、MODIFY_ACL、または OWNER
provider	オプション	文字列	ユーザーへのセキュリティの適用に使用するセキュリティプロバイダ (Native など) です。	ネイティブ
cascade	オプション	ブール	セキュリティ設定を、指定したフォルダ内のすべてのファイルとサブフォルダに適用するかどうかを指定します。	True または False

テーブル 3-5  
applySecurity の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-6  
applySecurity の例外

種類	説明
ResourceNotFoundException	ソース ファイルは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	指定したユーザーまたはセキュリティプロバイダの名前が不正です。

## 例

次の例では、指定したファイルの READ 権限をユーザー icrod に割り当てます。

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

## cascadeSecurity メソッド

```
cascadeSecurity(source)
```

フォルダのセキュリティ設定を、このフォルダ内のすべてのファイルとサブフォルダに適用します。

テーブル 3-7  
cascadeSecurity の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder" または "0a58c3670016a7860000010dcee0eaa28219"

テーブル 3-8  
cascadeSecurity の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-9  
cascadeSecurity の例外

種類	説明
ResourceNotFoundException	ソース フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、フォルダ Projects のセキュリティをフォルダのすべての子にカスケードします。

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

## copyResource メソッド

copyResource(source, target)

ファイルまたはフォルダをリポジトリ内の他のフォルダにコピーします。指定したソース ファイルまたはフォルダをコピーするときに、名前を変更できます。名前の変更の詳細は、「[moveResource メソッド](#)」(p. 51) を参照してください。

テーブル 3-10  
copyResource の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"
target	必須	文字列	ファイルのコピー先フォルダの完全修飾パスまたはオブジェクト URI です。指定したファイルまたはフォルダをコピーするときに、新しい名前に変更できます。	"/New Folder" または "/New Folder/abc.dat"

テーブル 3-11  
copyResource の戻り値

種類	説明
文字列	コピーされるファイルまたはフォルダの URI です。

テーブル 3-12  
copyResource の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、Drafts フォルダを Projects という名前のフォルダにコピーします。

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

## createFolder メソッド

```
createFolder(source)
```

リポジトリ内の指定した場所に新しいフォルダを作成します。

テーブル 3-13  
createFolder の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリで作成するフォルダ	"/Temp Folder/Sample Folder"

テーブル 3-14  
createFolder の戻り値

種類	説明
文字列	作成されたフォルダの URI です。

テーブル 3-15  
createFolder の例外

種類	説明
InsufficientParameterException	必須パラメータは指定されません。
ResourceAlreadyExistsException	指定したフォルダがリポジトリ内に既に存在しています。

## 例

次の例では、Drafts という名前のフォルダを Demo フォルダの子として作成します。フォルダ作成の問題が発生した場合、例外メッセージがコンソールに送信されます。

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

## deleteFile メソッド

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

リポジトリからファイルを削除します。ファイルのすべてのバージョンが削除されます。

テーブル 3-16  
deleteFile の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"
submittedHierarchy	オプション	ブール	ファイルが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-17  
deleteFile の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-18  
deleteFile の例外

種類	説明
ResourceNotFoundException	ソース ファイルは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	削除するよう指定されたリソースがフォルダです。

## 例

次の例では、ファイル MyReport.rptdesign をリポジトリから削除します。

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

## deleteFileVersion メソッド

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

リポジトリから、ファイルの特定のバージョンを削除します。

テーブル 3-19  
deleteFileVersion の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a78600 00010dcee0eaa28219"

フィールド	使用	種類	説明	値の例
version	オプション。ただし、version または label のいずれかを指定する必要があります。	文字列	削除するファイルの特定のバージョンです。	"0:2006-08-25 21:15:49.453"
label	オプション。ただし、version または label のいずれかを指定する必要があります。	文字列	削除するファイルの特定バージョンのラベルです。	"Version 1"
submittedHierarchy	オプション	ブール	ファイルが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-20  
deleteFileVersion の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-21  
deleteFileVersion の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	削除するよう指定されたリソースがフォルダです。

## 例

次の例では、Test というラベルのファイル MyReport.rptdesign をリポジトリから削除します。

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

## deleteFolder メソッド

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

リポジトリからフォルダとフォルダ内のすべてのコンテンツを削除します。

テーブル 3-22

deleteFolder の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder" または "0a58c3670016a78600 00010dcee0eaa28219"
submittedHierarchy	オプション	ブール	フォルダが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-23

deleteFolder の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-24

deleteFolder の例外

種類	説明
ResourceNotFoundException	指定したフォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	削除するよう指定されたリソースがフォルダではありません。

## 例

次の例では、Drafts という名前のファイルをリポジトリから削除します。フォルダ削除の問題が発生した場合、例外メッセージがコンソールに送信されます。

```

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."

```

## downloadFile メソッド

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

リポジトリから、特定のバージョンのファイルをローカル ファイル システムにダウンロードします。

テーブル 3-25  
downloadFile の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	ダウンロードするファイルの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"
target	必須	文字列	ファイルのダウンロード先フォルダの完全修飾パス (ローカル ファイル システム上) です。	"C:\Temp"
version	オプション。バージョンまたはラベルのいずれか一方を指定できます。	文字列	ダウンロードするファイルの特定のバージョンです。	"0:2006-08-25 21:15:49.453"
label	オプション。バージョンまたはラベルのいずれか一方を指定できます。	文字列	ダウンロードするファイルの特定バージョンのラベルです。	"Version 2"

テーブル 3-26  
downloadFile の戻り値

種類	説明
Resource	リポジトリ オブジェクトに関する情報のコンテナ。詳細は、p. 60 <a href="#">Resource クラス</a> を参照してください。

テーブル 3-27  
downloadFile の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、ファイル MyReport.rptdesign の Production というラベルのバージョンをローカル ファイル システムの Shared ディレクトリにダウンロードします。

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

## exportResource メソッド

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

指定されたリポジトリ フォルダをローカル ファイル システムの指定した \* .pes エクスポート ファイルにエクスポートします。

テーブル 3-28  
exportResource の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	エクスポートするフォルダの完全修飾リポジトリパスまたはオブジェクト URI です。	"/Temp Folder" または "0a58c3670016a78 60000010dcee0eaa2 8219"
target	必須	文字列	フォルダのエクスポート先として、ローカル ファイル システム上の完全修飾パスと、ファイルの名前を指定します。	"C:\Temp\backup.pes"

テーブル 3-29  
exportResource の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-30  
exportResource の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	指定したターゲットがフォルダです。ターゲットは *.pes ファイルである必要があります。

## 例

次の例では、Drafts フォルダの内容を、ローカル ファイル システムの backups フォルダのエクスポート ファイルにエクスポートします。

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

## getAccessControlList メソッド

getAccessControlList(source,submittedHierarchy)

リポジトリ内の指定したファイルまたはフォルダのセキュリティ アクセス コントロール リスト (ACL) を取得します。

テーブル 3-31  
getAccessControlList の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a78600 00010dcee0eaa28219"
submittedHierarchy	オプション	ブール	ファイルまたはフォルダが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-32  
getAccessControlList の戻り値

種類	説明
Dictionary	ユーザー名および関連付けられたアクセス許可を含む辞書が表示されます。次に例を示します。  { "admin": "MODIFY_ACL", "Joe": "DELETE" }

テーブル 3-33  
getAccessControlList の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、ファイル MyReport.rptdesign の ACL を印刷します。

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

## getAllVersions メソッド

getAllVersions(source, submittedHierarchy)

リポジトリ内のファイルのすべてのバージョンのリストを取得します。

テーブル 3-34  
getAllVersions の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a78600 00010dcee0ea28219"
submittedHierarchy	オプション	ブール	ファイルが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-35  
getAllVersions の戻り値

種類	説明
List	リソース オブジェクトの一覧です。「Resource クラス」 ( p. 60 ) を参照してください。

テーブル 3-36  
getAllVersions の例外

種類	説明
ResourceNotFoundException	ソース ファイルは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	指定したソースがフォルダです。

## 例

この例では、それぞれの作者、バージョン マーカー、バージョン ラベルを印刷し、すべてのバージョンのファイル MyReport.rptdesign に関する情報を取得します。

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

## getChildren メソッド

getChildren(source, submittedHierarchy)

リポジトリの指定したフォルダ内のすべてのファイルおよびフォルダのリストを取得します。

テーブル 3-37  
getChildren の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder" または "0a58c3670016a786000010dcee0eaa28219"
submittedHierarchy	オプション	ブール	フォルダが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-38  
getChildren の戻り値

種類	説明
List	リソース オブジェクトの一覧です。「Resource クラス」(p. 60) を参照してください。

テーブル 3-39  
getChildren の例外

種類	説明
InsufficientParameterException	必須パラメータは指定されません。
ResourceNotFoundException	フォルダは存在しません。

## 例

次の例では、それぞれのタイトル、著者、リソース ID を印刷し、/Demo/Drafts フォルダの子を取得します。

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

## getCustomPropertyValue メソッド

```
getCustomPropertyValue(propertyName)
```

指定したカスタム プロパティで使用できる有効な値を取得します。

テーブル 3-40  
getCustomPropertyValue の入力パラメータ

フィールド	使用	種類	説明	値の例
propertyName	必須	文字列	カスタム プロパティの名前です。	"FreeForm"

テーブル 3-41  
getCustomPropertyValue の戻り値

種類	説明
List	カスタム プロパティで使用できる有効な値のリストが返されます。プロパティで選択が必要な場合（単一選択、複数選択など）、リストにはその選択に対して有効な値がすべて含まれます。自由形式のプロパティの場合、リストにはプロパティで使用できるデータ型（String、Date、Number など）が含まれます。

テーブル 3-42  
getCustomPropertyValue の例外

種類	説明
ResourceNotFoundException	指定したプロパティは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次のサンプルは、カスタム プロパティ Language の値を使用します。

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

## getMetadata メソッド

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

カスタム プロパティおよびトピック情報も含めて、リポジトリ内のファイルまたはフォルダのメタデータ属性を取得します。

テーブル 3-43  
getMetadata の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a78600 00010dcee0eaa28219"
version	オプション。 バージョンまたはラベルのいずれか一方を指定できます。	文字列	ファイルまたはフォルダの特定のバージョンです。	"0:2006-08-25 21:15:49.453"
label	オプション。 バージョンまたはラベルのいずれか一方を指定できます。	文字列	ファイルまたはフォルダの特定バージョンのラベルです。	"Version 1"
submittedHierarchy	オプション	ブール	ファイルが [送信済みジョブ] フォルダにあるかどうかを指定します。	True または False

テーブル 3-44  
getMetadata の戻り値

種類	説明
Resource	リポジトリ オブジェクトに関する情報のコンテナ。詳細は、p. 60 <a href="#">Resource クラス</a> を参照してください。

テーブル 3-45  
getMetadata の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはフォルダが存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、/Demo/Drafts フォルダのリソース ID にアクセスします。

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

## importResource メソッド

importResource(source, target)

既存の \*.pes エクスポート ファイルをローカル ファイル システムからリポジトリにインポートします。

テーブル 3-46  
importResource の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	インポートするファイルのローカル ファイル システム上の完全修飾パスです。	"C:\Temp\New.pes"
target	必須	文字列	インポート先のフォルダの完全修飾リポジトリパスまたはオブジェクト URI です。	"/Temp Folder" または "0a58c3670016a7860000010dcee0eaa28219"

テーブル 3-47  
importResource の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-48  
importResource の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次に例を示します。

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

## moveResource メソッド

moveResource(source, target)

ファイルまたはフォルダをリポジトリ内の他のフォルダに移動します。ターゲットの種類およびファイルの有無で最終的な名前を決定し、移動するときに指定したソース ファイルの名前を決定できます。次の表は、ファイル移動時の名前変更機能の動作について示します。

テーブル 3-49  
ファイル名変更

ターゲットの種類	ターゲット フォルダが存在する	ターゲット フォルダが存在しない
フォルダ	ソース ファイルがターゲット フォルダの子となります。	ソース ファイルは指定したターゲット フォルダの親フォルダに移動し、ファイル名はターゲット フォルダの名前に変更されます。
ファイル	ソース ファイルは、ターゲット ファイルを含むフォルダに移動し、名前をターゲットの名前に変更します。	エラーが報告されます。

たとえば、ソースがファイル /Temp Folder/Temp.txt であり、指定したターゲットがフォルダ /Demo Folder である場合、結果が次のようになります。

- フォルダ Demo Folder が存在する場合、Temp.txt は Demo Folder に移動します。
- フォルダ Demo Folder が存在しない場合、Temp.txt は “ / “ に移動し、名前が Demo Folder に変更されます。

また、ソースがファイル /Temp Folder/Temp.txt であり、指定したターゲットがファイル /Demo Folder/Abc.dat である場合、結果が次のようになります。

- フォルダ Demo Folder が存在する場合、Temp.txt は Demo Folder に移動し、名前が Abc.dat に変更されます。
- フォルダ Demo Folder が存在しない場合、エラーが表示されます。

テーブル 3-50

moveResource の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a786000010dcee0eaa28219"
target	必須	文字列	ファイルの移動先フォルダの完全修飾パスまたはオブジェクト URI です。指定したファイルまたはフォルダを移動するときに、新しい名前に変更できます。	"/New Folder" または "/New Folder/abc.dat"

テーブル 3-51

moveResource の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-52

moveResource の例外

種類	説明
ResourceNotFoundException	指定したソースが存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次に例を示します。

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
```

```
bSuccess = pesImpl.moveResource(source="/Demo/Drafts/MyReport.rptdesign",
target="/Approved")
print bSuccess
```

## removeLabel メソッド

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

リポジトリ内のファイルからラベルを削除します。

テーブル 3-53

removeLabel の入力パラメータ

フィールド	使用	種類	値の例	説明
source	必須	文字列	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"	リポジトリ内のファイル の完全修飾パスまたはオ ブジェクト URI です。
label	必須	文字列	"Version 1"	削除するラベル名です。

テーブル 3-54

removeLabel の戻り値

種類	説明
文字列	更新されたファイルの URI です。

テーブル 3-55

removeLabel の例外

種類	説明
ResourceNotFoundException	ソース ファイルは存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、ファイル MyReport.rptdesign からラベル Draft を削除します。

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

## removeSecurity メソッド

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

リポジトリ内の指定したファイルまたはフォルダからセキュリティ アクセス コントロール リスト (ACL) を削除します。

テーブル 3-56  
removeSecurity の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"
principal	必須	文字列	指定したファイルまたはフォルダから削除するユーザー (admin など) です。	admin
provider	オプション	文字列	ユーザー情報の取得に使用するセキュリティプロバイダ (Native など) です。	ネイティブ
cascade	オプション	ブール	セキュリティ設定を、指定したフォルダ内のすべてのファイルとサブフォルダに適用するかどうかを指定します。	True または False

テーブル 3-57  
removeSecurity の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

テーブル 3-58  
removeSecurity の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
InsufficientParameterException	必須パラメータは指定されません。
IllegalParameterException	指定したユーザーまたはセキュリティプロバイダの名前が不正です。

## 例

次の例では、ファイル MyReport.rptdesign からプリンシパル icrod の ACL を削除します。

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

## search メソッド

```
search(criteria)
```

リポジトリのファイルを検索し、検索基準に一致するメタデータ コンテンツを含むファイル バージョンの一覧を返します。

テーブル 3-59  
search の入力パラメータ

フィールド	使用	種類	説明	値の例
criteria	必須	文字列	ファイルのメタデータを検索するための値です。	"Age"

テーブル 3-60  
search の戻り値

種類	説明
PageResult	行が検索の一致に対応する構造。詳細は、 p. 62 <a href="#">PageResult クラス</a> を参照してください。

テーブル 3-61  
search の例外

種類	説明
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、メタデータ フィールドにテキスト Quarterly があるファイル バージョンを検索します。

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

## setLabel メソッド

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

リポジトリ内の任意のバージョンのファイルにラベルを適用します。ファイルに既にラベルが付いている場合は、元のラベルが新しいラベルに置き換えられます。

テーブル 3-62  
setLabel の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt" または "0a58c3670016a7860000 010dcee0eaa28219"
version	必須	文字列	ファイルの特定のバージョンです。	"0:2006-08-25 21:15:49.453"
label	必須	文字列	ファイルに適用するラベルです。	"Version 1"

テーブル 3-63  
setLabel の戻り値

種類	説明
文字列	更新されたファイルの URI です。

テーブル 3-64  
setLabel の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはバージョンが存在しません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

次の例では、ラベル Beta を 2 番目のバージョンのファイル MyReport.rptdesign に割り当てます。Resource オブジェクトの getVersionMarker メソッドが、ラベルづけする必要のあるバージョンのマーカールを返します。

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

## setMetadata メソッド

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

メタデータ プロパティをリポジトリのファイルとフォルダに適用します。次の表は、メタデータ プロパティとファイルおよびフォルダのどちら（または両方）に適用されるかを示しています。

テーブル 3-65  
リポジトリ オブジェクトのプロパティ

メタデータのプロパティ	リソースのタイプ
著者	ファイル
説明	ファイルまたはフォルダ
タイトル	ファイルまたはフォルダ
有効期限	ファイルまたはフォルダ
キーワード	ファイル
トピック	ファイル
カスタム メタデータ	ファイルまたはフォルダ

テーブル 3-66  
setMetadata の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のファイルまたはフォルダの完全修飾パスまたはオブジェクト URI です。	"/Temp Folder/Temp.txt " または "0a58c3670016a7860000 010dcee0eaa28219"
version	オプション。バージョンまたはラベルのいずれか一方を指定できます。	文字列	ダウンロードされるファイルの特定のバージョンです。	"0:2006-08-25 21:15:49.453"

フィールド	使用	種類	説明	値の例
label	オプション。バージョンまたはラベルのいずれか一方を指定できます。	文字列	特定のバージョンのラベルです。	"Label 1"
props	必須	Dictionary	設定されるメタデータがすべて含まれ、辞書にはメタデータ名がキーとして含まれます。「値の例」に示すように、値として topic のリストと customProperty の辞書を受け取ります。これ以外のメタデータについては、文字列を受け取ります。	{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }

テーブル 3-67  
setMetadata の戻り値

種類	説明
文字列	メタデータの設定対象となったファイルまたはフォルダの URI です。

テーブル 3-68  
setMetadata の例外

種類	説明
InsufficientParameterException	必須パラメータは指定されません。
ResourceNotFoundException	ソース ファイルまたはフォルダが存在しません。

## 例

次の例では、キーワード Quarterly を Production のバージョンのファイル MyReport.rptdesign に割り当てます。

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

## uploadFile メソッド

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

ファイルをローカル ファイル システムからリポジトリに保存します。ファイルが既に存在する場合は、オプションとして新しいバージョンのファイルを作成できます。

テーブル 3-69  
uploadFile の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	アップロードするファイルのローカル ファイル システム上の完全修飾パスです。	"C:\Temp\Temp.txt"
target	必須	文字列	リポジトリ内のファイルのアップロード先となるフォルダの完全修飾パスです。	"/Temp Folder"
versionFlag	オプション	ブール	指定したファイルが存在する場合、新しいバージョンのファイルが作成されます。	True または False

テーブル 3-70  
uploadFile の戻り値

種類	説明
文字列	アップロードされたファイルの URI です。

テーブル 3-71  
uploadFile の例外

種類	説明
ResourceNotFoundException	ソース ファイルまたはターゲット フォルダは存在しません。
ResourceAlreadyExistsException	この例外は、ターゲット フォルダにソース ファイルと同じ名前のファイルまたはフォルダが存在し、createVersion パラメータが指定されていません。
InsufficientParameterException	必須パラメータは指定されません。

## 例

この例では、ファイル MyReport.rptdesign をリポジトリの /Demo/Drafts フォルダにアップロードします。ファイルが既に存在する場合、新しいバージョンのファイルが versionFlag パラメータを使用してアップロードされます。

```

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 クラス

PESImpl API には、コンテンツ リポジトリ メソッドによって呼び出される Web サービスから返されるオブジェクトのラッパーとして機能するクラスが含まれています。これらのラッパー クラスには、メソッドによって返される情報を表示するためのインターフェイスがあります。

## Resource クラス

Resource クラスは、オブジェクト固有の情報に対するアクセスを提供し、リポジトリ オブジェクト ResourceSpecifer.Resource への単純化されたラッパーとして機能します。リポジトリ オブジェクトに関連する標準メタデータのほか、このクラスにはリポジトリのオブジェクトに定義されたカスタム メタデータ情報が含まれます。[テーブル 3-72](#) は、Resource クラスで使用できるすべてのメソッドです。

テーブル 3-72  
Resource クラスのメソッド

メソッド名	説明
getAccessControlList	オブジェクトのセキュリティ許可の辞書を返します。辞書には、キーとしてユーザー名と、そのユーザーに付与された最上位の許可が格納されます。次に例を示します。 ユーザー Joe が resource X の削除権限を割り当てられている場合、X を表すリソース オブジェクトの getAccessControlList は、Web サービスの呼び出しから 3 つの権限（読み取り、書き込み、削除）すべてではなく {'Joe': 'DELETE'} を返します。
getOwner	オブジェクトの所有者の名前を文字列として返します。
getAuthor	オブジェクトの著者の名前を文字列として返します。
getContentSize	オブジェクトのサイズを返します。
getCreatedBy	オブジェクトを作成したユーザーの名前を文字列として返します。
getCreationDate	オブジェクトの作成日を日付時刻型オブジェクトとして返します。
getDescription	オブジェクトの説明をリストとして返します。

メソッド名	説明
getDescriptionLanguage	オブジェクトの言語をリストとして返します。
getExpirationDate	オブジェクトの有効期限を日付時刻型オブジェクトとして返します。
isExpired	指定したオブジェクトが期限切れかどうかを示します。
getMimeType	オブジェクトの MIME タイプを文字列として返します。
getModificationDate	オブジェクトの最終更新日を日付時刻型オブジェクトとして返します。
getObjectCreationDate	オブジェクトのオブジェクト作成日を日付時刻型オブジェクトとして返します。
getObjectLastModifiedBy	オブジェクトを最終更新したユーザーを文字列として返します。
getObjectLastModifiedDate	オブジェクトのオブジェクト最終更新日を日付時刻型オブジェクトとして返します。
getResourceID	オブジェクトのリソース識別子を文字列として返します。
getResourcePath	指定したオブジェクトのパスを文字列として返します。
getTitle	オブジェクトのタイトルを文字列として返します。
getTopicList	オブジェクトのトピック リストを返します。
getVersionMarker	オブジェクトのバージョンを文字列として返します。
getVersionLabel	オブジェクトのラベルを文字列として返します。
getCustomMetadata	オブジェクトに関連付けられたカスタム プロパティを辞書として返します。
getKeywordList	オブジェクトに関連付けられたキーワードのリストを返します。

## IdentificationSpecifier クラス

このクラスは、オブジェクトの ID 固有のデータへのアクセスを可能にするコンテンツ リポジトリ オブジェクト IdentificationSpecifier として機能します。 [テーブル 3-73](#) は、IdentificationSpecifier クラスで使用できるすべてのメソッドです。

テーブル 3-73  
IdentificationSpecifier クラスのメソッド

メソッド名	説明
getIdentifier	オブジェクトの ID 値を文字列として返します。

メソッド名	説明
getVersionMarker	オブジェクトのバージョンを文字列として返します。
getVersionLabel	オブジェクトバージョンに適用されたラベルを文字列として返します。

## PageResult クラス

PageResult クラスは、検索結果のコンテナとして機能します。結果の個別のヒットは、PageResult オブジェクトの行に対応します。たとえば、4 つのリソースを返す検索は、4 つの行を含む PageResult オブジェクトを生成します。[テーブル 3-74](#) は、PageResult クラスで使用できるすべてのメソッドです。

テーブル 3-74  
PageResult クラスのメソッド

メソッド名	説明
getRows	SearchRow オブジェクトのリストを返します。詳細は、p. 62 <a href="#">SearchRow クラス</a> を参照してください。

## SearchRow クラス

SearchRow クラスは、個々の検索結果に関するオブジェクトレベルの情報のコンテナとして機能します。このクラスのメソッドを使用して、オブジェクトに関するメタデータにアクセスできます。[テーブル 3-75](#) は、SearchRow クラスで使用できるすべてのメソッドです。

テーブル 3-75  
SearchRow クラスのメソッド

メソッド名	説明
getTitle	ファイルまたはフォルダの名前を返します。
getAuthor	ファイルまたはフォルダの著者を返します。
getMimeType	ファイルの MIME タイプを返します。
getObjectLastModifiedBy	ファイルまたはフォルダを最終更新したユーザーを返します。
getModified	ファイルまたはフォルダが最終更新された日付と時刻を返します。
getFolderPath	ファイルまたはフォルダの場所を返します。
getFolder	ファイルまたはフォルダの親フォルダの名前を返します。
getParentURI	親のオブジェクト URI を返します。

メソッド名	説明
getTopic	ファイルまたはフォルダに関連付けられたトピックを返します。
getChildRow	SearchChildRow オブジェクト (詳細は次の項を参照) のリストを返します。

オブジェクトのバージョン レベルの情報にアクセスするには、getChildRow メソッドを使用して、オブジェクト バージョンに対応する子行を返します。

## SearchChildRow クラス

SearchChildRow クラスは、個々の検索結果に関するバージョンレベルの情報のコンテナとして機能します。このクラスのメソッドを使用して、オブジェクト バージョンに関するメタデータにアクセスできます。 [テーブル 3-76](#) は、SearchChildRow クラスで使用できるすべてのメソッドです。

テーブル 3-76  
SearchChildRow クラスのメソッド

メソッド名	説明
getExpirationDate	ファイルまたはフォルダの有効期限を返します。
getKeyword	ファイルまたはフォルダのバージョンに関連付けられたキーワードを返します。
getVersionLabel	ファイルまたはフォルダのバージョン ラベルを返します。
getDescription	ファイルまたはフォルダの説明を返します。
getLanguage	言語を返します。
getVersionCreationDate	ファイルまたはフォルダが作成された日付と時刻を返します。
getVersionMarker	ファイルまたはフォルダのバージョン マーカーを返します。
getUri	ファイルまたはフォルダのオブジェクト URI を返します。

## プロセス管理 API

プロセス管理スクリプトには、ジョブを処理する機能があります。この領域には次の機能があります。

- ジョブの実行
- ジョブ履歴の取得
- ジョブ詳細の取得

この項では、リポジトリに保存されたジョブの操作に使用する PESImpl メソッドについて概要を説明します。すべてのメソッドには、詳細なシNTAX情報、例、予測されるメッセージが含まれています。

## メソッド

次の項では、IBM® SPSS® Collaboration and Deployment Services で使用できるすべてのプロセス管理スクリプトメソッドのリストです。

注: リポジトリのファイルまたはフォルダへのパスを必要とするすべてのメソッドについては、パスまたはオブジェクト URI のいずれかを使用できます。オブジェクト URI は、IBM® SPSS® Collaboration and Deployment Services Deployment Manager でオブジェクトプロパティを表示することにより取得できます。

### cancelJob メソッド

```
cancelJob(executionId)
```

実行中のジョブをキャンセルします。

テーブル 3-77

cancelJob の入力パラメータ

フィールド	使用	種類	説明	値の例
executionId	必須	文字列	ジョブの実行 ID です。	0a58c33d002ce90800 00010e0ccf7b01800e

テーブル 3-78

cancelJob の戻り値

種類	説明
ブール	ジョブがキャンセルされると成功メッセージが返されます。

### 例

この例は、レポートジョブの実行を終了します。

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

### deleteJobExecutions メソッド

```
deleteJobExecutions(executionId)
```

1 つまたは複数のジョブ実行を削除します。

テーブル 3-79

deleteJobExecutions の入力パラメータ

フィールド	使用	種類	説明	値の例
executionId	必須	List	ジョブ実行識別子のリスト	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

テーブル 3-80

deleteJobExecutions の戻り値

種類	説明
ブール	True または False。メソッドが正常に実行されたかどうかによって決まります。

## 例

この例は、レポート ジョブの実行を削除します。

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

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

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

## executeJob メソッド

executeJob(source, 通知, asynchronous)

指定したパラメータに基づき、ジョブを同期または非同期で実行します。同期で実行する場合は、ジョブが完了するまでメソッドは返されません。非同期で実行する場合は、ジョブが開始した後にメソッドは応答します。

テーブル 3-81

executeJob の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	アップロードするファイルのローカルファイルシステム上の完全修飾パスです。	"C:\Temp\Temp.txt"

フィールド	使用	種類	説明	値の例
通知	オプション	ブール	ジョブの実行を通知するかどうかを指定します。デフォルトは False です。	True または False
asynchronous	オプション	ブール	ジョブを非同期で実行するかどうかを指定します。デフォルトは False です。	True または False

テーブル 3-82  
executeJob の戻り値

種類	説明
文字列	ジョブ実行の一意の識別子。識別子を使用して、特定のジョブ実行を参照します。

## 例

この例は、レポート ジョブの実行を通知とは非同期で開始します。

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

## getJobExecutionDetails メソッド

getJobExecutionDetails(executionId, ログ, target)

すべてのジョブ ステップと繰り返しを含めて、特定ジョブの実行の詳細情報を一覧表示します。

テーブル 3-83  
getJobExecutionDetails の入力パラメータ

フィールド	使用	種類	説明	値の例
executionId	必須	文字列	ジョブの実行 ID です。	0a58c33d002ce9080000 010e0ccf7b01800e
ログ	オプション	ブール	ジョブのログをインラインで表示するかどうかを指定します。	True または False
target	オプション	文字列	ログを保存するローカルファイル システム上の場所です。log パラメータを指定する場合にのみ併用します。	"c:\logs"

テーブル 3-84  
getJobExecutionDetails の戻り値

種類	説明
jobExecutionDetails	ジョブ実行に関する詳細。詳細は、p. 69 <a href="#">jobExecutionDetails クラス</a> を参照してください。

### 例

この例は、識別子が execId であるジョブ実行のジョブ ステップ実行に関する情報を取得し、各ステップの結果をコンソールに送信します。

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

## getJobExecutionList メソッド

getJobExecutionList(source)

実行中のジョブと完了済みのジョブについて、特定のジョブの全バージョンを対象にしてその実行を一覧表示します。

テーブル 3-85  
getJobExecutionList の入力パラメータ

フィールド	使用	種類	説明	値の例
source	必須	文字列	リポジトリ内のジョブの完全修飾パスです。	"/testJob"

テーブル 3-86  
getJobExecutionList の戻り値

種類	説明
PageResult	ジョブの実行のリストのコンテナ。詳細は、p. 68 <a href="#">PageResult クラス</a> を参照してください。

### 例

この例は、レポート ジョブ実行を取得し、各ステップ実行に関する情報をコンソールに送信します。

```
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 クラス

PESImpl API には、プロセス管理メソッドによって呼び出される Web サービスから返されるオブジェクトのラッパーとして機能するクラスが含まれています。これらのラッパー クラスには、メソッドによって返される情報を表示するためのインターフェイスがあります。

### PageResult クラス

PageResult クラスは、検索結果のコンテナとして機能します。ジョブ実行に固有のデータを取得できます。個別のジョブ実行は、PageResult オブジェクトの行に対応します。たとえば、4 回実行されたジョブは 4 行を含む PageResult オブジェクトに対応します。テーブル 3-87 は、PageResult クラスで使用できるすべてのメソッドです。

テーブル 3-87  
PageResult クラスのメソッド

メソッド名	説明
getRows	それぞれがジョブの実行を示す Row オブジェクトのリストを返します。詳細は、p. 68 <a href="#">Row クラス</a> を参照してください。

### Row クラス

Row クラスは、ジョブ実行に関するジョブレベルの情報のコンテナとして機能します。このクラスのメソッドを使用して、ジョブ実行に関するメタデータにアクセスできます。テーブル 3-88 は、Row クラスで使用できるすべてのメソッドです。

テーブル 3-88  
Row クラスのメソッド

メソッド名	説明
getObjId	ジョブの実行 ID を返します。
getPath	ジョブのパスを返します。
getVersionMarker	実行されたジョブのバージョン マーカーを返します。

メソッド名	説明
getVersionLabel	実行されたジョブのバージョン ラベルを返します。
getEventObjId	実行されたジョブのイベント ID を返します。
getEventState	実行中のジョブの状態を返します。
getEventCompletionCode	ジョブの完了コードを返します。
getEventStartDateTime	ジョブが開始された日付と時刻を返します。
getEventEndDateTime	ジョブが終了した日付と時刻を返します。
getQueuedDateTime	ジョブがキューに登録された日付と時刻を返します。

## jobExecutionDetails クラス

このクラスは、getJobExecutionDetails メソッドから返されます。ジョブ実行の詳細情報が保存され、ジョブの各ステップに関する情報を示す jobStepExecution オブジェクトのリストが含まれます。[テーブル 3-89](#) は、jobExecutionDetails クラスで使用できるすべてのメソッドです。

テーブル 3-89  
jobExecutionDetails クラスのメソッド

メソッド名	説明
getJobStepDetails	jobStepExecutionDetails オブジェクトのリストを返します。詳細は、 <a href="#">p. 70 jobStepExecutionDetails クラス</a> を参照してください。
getArtifactLocation	ジョブのアーティファクトの場所を返します。
getCompletionCode	ジョブ実行の完了コードを返します。
getEndDateTime	ジョブ実行が終了した日付と時刻を返します。
getEventName	ジョブ実行のイベント名を返します。
getEventUUID	ジョブ実行のイベント ID を返します。
getExecutionState	ジョブ実行の実行状態を返します。
getExecutionSuccess	ジョブ実行の成功または失敗のステータスを返します。
getExecutionWarning	警告が発生したかどうかを示します。

メソッド名	説明
getLog	生成されたログを文字列として返します。
getNotificationEnabled	Eメール通知が有効になっているかどうかを示します。
getQueuedDateTime	ジョブ実行がキューに登録された日付と時刻を返します。
getStartDateTime	ジョブ実行が開始された日付と時刻を返します。
getUserName	ジョブを実行したユーザーの名前を返します。
getUUID	ジョブの実行 ID を返します。

### jobStepExecutionDetails クラス

このクラスには、ジョブ ステップ実行の詳細情報と、`jobStepChildExecutionDetails` オブジェクトのリストが保存されます。このクラスには、すべてのメソッド呼び出しが委任される `ExecutionDetails` オブジェクトが含まれます。[テーブル 3-90](#) は、`jobStepExecutionDetails` クラスで使用できるすべてのメソッドです。

テーブル 3-90

#### jobStepExecutionDetails クラスのメソッド

メソッド名	説明
getJobStepChldExecution-List	<code>jobStepChildExecutionDetails</code> オブジェクトのリストを返します。詳細は、 <a href="#">p. 71 jobStepChildExecutionDetails クラス</a> を参照してください。
getArtifactLocation	ジョブ ステップのアーティファクトの場所を返します。
getCompletionCode	ジョブ ステップの完了コードを返します。
getEndDateTime	ジョブ ステップが終了した日付と時刻を返します。
getEventName	ジョブ ステップのイベント名を返します。
getEventUUID	ジョブ ステップのイベント ID を返します。
getExecutionState	ジョブ ステップの実行状態を返します。
getExecutionSuccess	ジョブ ステップの成功または失敗のステータスを返します。
getExecutionWarning	警告が発生したかどうかを示します。
getLog	生成されたログを文字列として返します。

メソッド名	説明
getNotificationEnabled	E メール通知が有効になっているかどうかを示します。
getQueuedDateTime	ジョブ ステップがキューに登録された日付と時刻を返します。
getStartDateTime	ジョブ ステップが開始された日付と時刻を返します。
getUserName	ジョブ ステップを実行したユーザーの名前を返します。
getUUID	ジョブ ステップの実行 ID を返します。

## jobStepChildExecutionDetails クラス

jobStepChildExecutionDetails クラスは、各ジョブ ステップの子実行のコンテナとして機能します。たとえば、反復するレポート ジョブ ステップは、ステップの各反復に子実行を生成します。このクラスのメソッドを使用して、子実行に関するメタデータにアクセスできます。[テーブル 3-91](#) は、jobStepChildExecutionDetails クラスで使用できるすべてのメソッドです。

テーブル 3-91  
jobStepChildExecutionDetails クラスのメソッド

メソッド名	説明
getArtifactLocation	子実行のアーティファクトの場所を返します。
getCompletionCode	子実行の完了コードを返します。
getEndDateTime	子実行が終了した日付と時刻を返します。
getEventName	子実行のイベント名を返します。
getEventUUID	子実行のイベント ID を返します。
getExecutionState	子実行の実行状態を返します。
getExecutionSuccess	子実行の成功または失敗のステータスを返します。
getExecutionWarning	警告が発生したかどうかを示します。
getLog	生成されたログを文字列として返します。
getNotificationEnabled	E メール通知が有効になっているかどうかを示します。
getQueuedDateTime	子実行の問い合わせが行われた日付と時刻を返します。
getStartDateTime	子実行が開始された日付と時刻を返します。

メソッド名	説明
getUserName	子実行を実行したユーザーの名前を返します。
getUUID	子実行の実行 ID を返します。

# 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](http://www.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.



# 索引

- ACL, 35, 45, 53
  - advanceSearch 操作, 7
  - advanceSearch メソッド, 33
  - applySecurity 操作, 9
  - applySecurity メソッド, 35
  - cancelJob メソッド, 64
  - cascadeSecurity 操作, 10
  - cascadeSecurity メソッド, 36
  - copyResource 操作, 10
  - copyResource メソッド, 37
  - createFolder 操作, 11
  - createFolder メソッド, 38
  - deleteFile 操作, 12
  - deleteFile メソッド, 39
  - deleteFileVersion 操作, 13
  - deleteFileVersion メソッド, 40
  - deleteFolder 操作, 14
  - deleteFolder メソッド, 42
  - deleteJobExecutions 操作, 28
  - deleteJobExecutions メソッド, 64
  - downloadFile 操作, 14
  - downloadFile メソッド, 43
  - executeJob 操作, 28
  - executeJob メソッド, 65
  - export 操作, 15
  - exportResource メソッド, 44
  - getAccessControllist 操作, 16
  - getAccessControllist メソッド, 45
  - getAllVersions 操作, 17
  - getAllVersions メソッド, 46
  - getChildren 操作, 17
  - getChildren メソッド, 47
  - getCustomPropertyValue 操作, 18
  - getCustomPropertyValue メソッド, 48
  - getJobExecutionDetails 操作, 29
  - getJobExecutionDetails メソッド, 66
  - getJobExecutionList 操作, 30
  - getJobExecutionList メソッド, 67
  - getMetadata 操作, 18
  - getMetadata メソッド, 49
  - IBM i, 3
  - IdentificationSpecifier クラス, 61
  - import 操作, 19
  - importResource メソッド, 50
  - jobExecutionDetails クラス, 69
  - jobStepChildExecutionDetails クラス, 71
  - jobStepExecutionDetails クラス, 70
  - legal notices, 73
  - moveResource 操作, 20
  - moveResource メソッド, 51
  - PageResult クラス, 34, 55, 62, 67–68
  - PESImpl モジュール, 32
  - removeLabel 操作, 21
  - removeLabel メソッド, 53
  - removeSecurity 操作, 22
  - removeSecurity メソッド, 53
  - Resource クラス, 44, 46–47, 50, 60
  - Row クラス, 68
  - search
    - 詳細, 33
  - search 操作, 23
  - search メソッド, 55
  - SearchChildRow クラス, 63
  - SearchRow クラス, 62
  - Secure Sockets Layer, 32
  - setLabel 操作, 23
  - setLabel メソッド, 55
  - setMetadata 操作, 24
  - setMetadata メソッド, 56
  - SSL, 32
  - trademarks, 74
  - UNIX, 2
  - uploadFile 操作, 26
  - uploadFile メソッド, 59
  - Windows, 2
- アクセス コントロール リスト  
適用, 35, 45, 53
- インストール, 1
  - IBM i, 3
  - UNIX, 2
  - Windows, 2
- ジョブ
  - 実行, 64–65, 67
  - キャンセル, 64
  - 実行の詳細, 66
- ファイル
  - 削除, 39, 51
  - アップロード, 59
  - インポート, 50

---

エクスポート, 44  
コピー, 37  
ダウンロード, 43  
バージョン, 46  
メタデータ, 49  
ファイルバージョン  
削除, 40  
フォルダ  
子, 47  
作成, 38, 42, 51  
コピー, 37  
メタデータ, 49  
  
ラベル  
削除, 53, 55