

Références pour IBM SPSS
Collaboration and Deployment
Services - Essentials for Python 5



Remarque : Avant d'utiliser ces informations et le produit associé, lisez les informations générales sous Remarques sur p. 65.

Cette version concerne IBM SPSS Collaboration and Deployment Services 5 et toutes les publications et modifications suivantes jusqu'à mention contraire dans les nouvelles versions.

Les captures d'écran des produits Adobe sont réimprimées avec l'autorisation d'Adobe Systems Incorporated.

Les captures d'écran des produits Microsoft sont réimprimées avec l'autorisation de Microsoft Corporation.

Matériel sous licence – Propriété d'IBM

© **Copyright IBM Corporation 2000, 2012.**

Restriction des droits des utilisateurs du gouvernement des États-Unis : Utilisation, duplication et divulgation sujettes à restriction de la part de GSA ADP Schedule Contract avec IBM Corp.

Préface

IBM® SPSS® Collaboration and Deployment Services est une application conçue pour l'entreprise, qui permet l'utilisation et le déploiement généralisés de l'analyse prédictive. IBM SPSS Collaboration and Deployment Services offre un stockage centralisé, sûr et vérifiable, des ressources analytiques, des capacités avancées de gestion et de contrôle des processus d'analyse prédictive, ainsi que des mécanismes perfectionnés de présentation des résultats du traitement analytique aux utilisateurs finaux. Parmi les avantages de l'application IBM SPSS Collaboration and Deployment Services, citons la protection de la valeur des ressources analytiques, l'assurance de conformité avec les exigences réglementaires, l'amélioration de la productivité des analystes et la réduction des coûts informatiques pour la gestion des analyses.

Assistance technique

Les services de l'assistance technique de IBM Corp. sont proposés aux clients enregistrés de IBM Corp.. Les clients peuvent contacter l'assistance technique pour obtenir de l'aide concernant l'utilisation des produits IBM Corp. ou l'installation dans l'un des environnements matériels pris en charge. Pour contacter le support technique, visitez le site Web de IBM Corp. à l'adresse <http://www.spss.com> ou contactez votre bureau local dont les coordonnées figurent sur le site Web de IBM Corp. à l'adresse : <http://www.spss.com/worldwide/>. Votre nom, celui de votre société, ainsi que le numéro de série de votre système, vous seront demandés.

Votre avis nous intéresse

Vos remarques sont importantes. Faites-nous part de votre expérience des produits IBM Corp.. Envoyez un e-mail à l'adresse suivante : suggest@us.ibm.com ou écrivez à SPSS Inc., Attn.: Director of Product Planning, 233 South Wacker Drive, 11th Floor, Chicago, IL 60606-6412, Etats-Unis.

Contenu

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

Présentation	1
Installation	1
Installation sous Windows	1
Installation sous UNIX	2
Installation sur IBM i	3

2 Scripts de ligne de commande **4**

Mots-clés globaux	4
Connexions du référentiel	5
Génération de scripts pour le référentiel de contenu	5
Mots clés	6
Opérations	7
Fonctions de gestion des processus	25
Mots clés	25
Opérations	26

3 Le module PESImpl **30**

API de Référentiel de contenu	30
Méthodes	30
Classes wrapper	53
API de gestion des processus	56
Méthodes	57
Classes wrapper	60
Exemples de scripts	64

Annexe

A Remarques

65

Index

68

IBM SPSS Collaboration and Deployment Services - Essentials for Python

Présentation

IBM® SPSS® Collaboration and Deployment Services fournit une structure de génération de scripts avec un ensemble d'API que les utilisateurs avancés et les administrateurs peuvent utiliser pour écrire des programmes indépendants ou des tâches par lots combinant un ensemble de programmes, à utiliser avec les tâches et objets du référentiel. Ceci permet de simplifier considérablement les tâches en masse telles que :

- La modification des droits d'accès de sécurité pour un grand groupe d'utilisateurs
- L'ajout ou la suppression d'une étiquette pour un grand nombre de dossiers ou de fichiers
- Le téléchargement ou le transfert d'un grand nombre de dossiers ou de fichiers

La structure inclut la possibilité d'effectuer des tâches depuis la ligne de commande, ainsi qu'une API complète pour l'interaction avec le IBM® SPSS® Collaboration and Deployment Services Repository dans votre propre code Python.

Pour obtenir des informations générales sur Python, un langage de programmation dynamique orienté objet, consultez le [site Python \(http://www.python.org\)](http://www.python.org).

Installation

La structure de génération de scripts peut être installée sur les plates-formes Windows, Unix et IBM i. La plateforme des scripts est indépendante de la plateforme utilisée par le référentiel accessible depuis l'utilitaire de scripts. Par exemple, un référentiel fonctionnant sur la plateforme Windows peut être appelé à l'aide des fonctions de scripts s'exécutant sur la plateforme UNIX.

Installation sous Windows

1. Si Python est déjà installé sur votre système, désinstallez-le.
2. Insérez le support d'installation.
3. Ouvrez le répertoire `PYTHON\Disk1\InstData\NoVM` du disque 2 et double-cliquez sur `install.exe`. Suivez les instructions à l'écran pour mener à terme l'installation. Installer à l'emplacement par défaut. Ceci installe les technologies Python, ZSI et PyXML nécessaires.
4. Ouvrez le répertoire `PYTHON` du support d'installation et extrayez le contenu de `cads-scripting-1.0.zip` dans un répertoire temporaire.

5. Ajoutez l'emplacement du répertoire de IBM® SPSS® Collaboration and Deployment Services - Essentials for Python à la variable d'environnement système du **chemin d'accès** de votre ordinateur.
6. À l'invite d'une commande, modifiez le répertoire actuel pour le dossier où vous avez extrait *cads-scripting-1.0.zip*. Saisissez la commande suivante et appuyez sur Entrée.

```
python setup.py install
```

Installation sous UNIX

1. Si Python 2.7.1, ZSI 2.0 rc3 et PyXML 0.8.4 ne sont pas déjà installés sur votre système, installez-les après les avoir téléchargés à partir de leurs sites Web respectifs avant de passer à l'étape 2.
 - Python 2.7.1: <http://www.python.org/download/releases/2.7.1/>
 - ZSI 2.0 rc3: <http://sourceforge.net/projects/pywebsvcs>
 - PyXML 0.8.4: http://sourceforge.net/project/showfiles.php?group_id=6473

2. Insérez le disque 2.
3. Copiez le fichier *install.jar* sur votre système.
4. A l'aide de Java runtime version 1.5 ou ultérieure, exécutez la commande suivante :

```
java -jar install.jar -i console
```

5. Dans le répertoire d'installation, modifiez *setup.cfg*. Vérifiez que la valeur de *<PythonInstallDir>* correspond au chemin d'installation de IBM® SPSS® Collaboration and Deployment Services - Essentials for Python. Si aucune valeur n'est spécifiée, le chemin sera défini par défaut sur la bibliothèque Python, par exemple */usr/lib/python2.7*.

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

6. À l'invite d'une commande, modifiez le répertoire actuel pour le dossier où vous avez extrait *cads-scripting-1.0.tar.gz*. Exécutez la commande suivante :

```
python setup.py install
```

Lors de l'extraction de groupes tar, certains utilitaires tar peuvent afficher un message similaire à celui-ci :

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

La valeur de ##### correspond à une valeur entière. Ces messages sont à but d'information uniquement et n'indiquent pas un échec de l'extraction.

Installation sur IBM i

1. Connectez-vous à votre système IBM i à l'aide d'un terminal Telnet.
2. Insérez le disque 1.
3. Démarrez QShell à l'aide de la commande suivante

```
QSH
```

4. Modifiez le répertoire en */qopt/Server/IBMi/Python*.
5. Exécutez la commande suivante :

```
java -jar install.jar -i console
```

Python est installé dans */QOpenSys/usr/local/bin/python2.7* et IBM® SPSS® Collaboration and Deployment Services - Essentials for Python est installé dans */QOpenSys/usr/local/lib/python2.7/site-packages*.

Scripts de ligne de commande

Le fichier Python *CADSTool.py* peut être utilisé à partir de la ligne de commande pour manipuler des ressources stockées dans le IBM® SPSS® Collaboration and Deployment Services Repository. La syntaxe générale utilisée pour appeler les opérations de génération de scripts IBM® SPSS® Collaboration and Deployment Services à partir de la ligne de commande est :

```
python CADSTool.py <Opération> <Mots clés>
```

où :

- <Opération> désigne la fonction à appeler
- <Keywords> (Mots clés) définit les paires valeur/mot clé utilisées comme paramètres d'entrée de la fonction

Mots-clés globaux

Table 2-1 répertorie les mots clés pris en charge par toutes les fonctions de génération de scripts IBM® SPSS® Collaboration and Deployment Services. La deuxième colonne décrit des versions abrégées facultatives des mots clés. Notez que les mots clés sont sensibles à la casse.

Table 2-1
Mots clés globaux

Mot clé	Version courte facultative	Utilisation
--user	-u	Le nom d'utilisateur servant à la connexion au serveur de référentiel
--password	-p	Le mot de passe servant à la connexion au serveur de référentiel
--host	-q	Le nom de l'hôte/du serveur sur lequel le référentiel est installé
--port	-o	Le numéro de port du serveur de référentiel
--useDefault	-z	Indique que les valeurs pour user, password, host et port doivent être lues à partir du fichier <i>Authorization.properties</i>
--ssl		Indique que le serveur de référentiel utilise le protocole secure sockets layer (SSL) pour chiffrer les communications. Avant d'utiliser ce mot clé, le serveur de référentiel doit être configuré pour le protocole SSL. Pour plus d'informations, reportez-vous à la documentation de l'administrateur.
-h		Les informations d'aide du module de génération de scripts

Connexions du référentiel

Vous devez indiquer l’ID d’utilisateur, le mot de passe, l’hôte et le port du IBM® SPSS® Collaboration and Deployment Services Repository à la fin de chaque commande. Les méthodes suivantes peuvent être utilisées pour fournir ces informations de connexion :

- Utilisation des mots-clés, tels que :

```
--user <utilisateur> --password <mot de passe> --host  
<hôte> --port <port>
```

- Via le fichier *Authorization.properties*, où la commande contient un paramètre `--useDefault` (ou la version courte `-z`). Ceci permet de récupérer les informations de connexion à partir du fichier *Authorization.properties* qui se trouve dans `<Scripting folder>\Lib\site-packages\config\Authorization.properties`. Un simple éditeur de texte permet de modifier les valeurs suivantes dans le fichier pour les faire correspondre aux paramètres de votre référentiel :

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

Les paramètres transmis via la ligne de commande sont toujours prioritaires. Par exemple, si `--user` et `--password` sont fournis via la ligne de commande et que le paramètre `--useDefault` ou `-z` est également indiqué, l’utilisateur et le mot de passe provenant de la ligne de commande sont utilisés, avec l’hôte et le port récupérés dans le fichier *Authorization.properties*. Ou, si l’utilisateur, le mot de passe, l’hôte et le port sont tous fournis via la ligne de commande mais que le paramètre `--useDefault` ou `-z` est également utilisé, `--useDefault` est ignoré et seules les informations de ligne de commande sont utilisées.

Pour toutes les API décrites ici, la syntaxe et les exemples utilisent le paramètre `-z` afin d’utiliser le nombre minimum de paramètres obligatoires.

Génération de scripts pour le référentiel de contenu

La génération de scripts du référentiel de contenu permet d’utiliser les ressources de référentiel, telles que les fichiers et les dossiers. Cette zone comprend les fonctionnalités suivantes :

- Création et suppression de dossiers
- Téléchargement et transfert de fichiers
- Exporter et importer des dossiers
- Gestion des étiquettes, de la sécurité et des métadonnées.

Cette section décrit l’utilisation de scripts de ligne de commande Python pour les fonctions du référentiel. Chaque opération contient des informations détaillées sur la syntaxe, un exemple et les messages attendus.

Mots clés

Table 2-2 répertorie les mots clés pris en charge pour les fonctions du référentiel. La deuxième colonne décrit des versions abrégées facultatives des mots clés.

Important : Les mots clés sont sensibles à la casse.

Table 2-2

Mots clés des API du référentiel

Mot clé	Version courte facultative	Utilisation
--source	-s	Le fichier source ou le chemin d'accès au dossier
--target	-t	Le chemin d'accès au dossier cible
--version	-v	La version d'un fichier
--principal	-r	L'utilisateur qui doit recevoir l'autorisation
--permission	-n	Le type d'autorisation (lecture, écriture, modification, suppression, par exemple)
--label	-l	L'étiquette à attribuer à une version d'un fichier
--criteria	-c	Les critères de recherche destinés à la recherche des attributs de métadonnées des fichiers ou des dossiers
--author	-a	Le nom de l'auteur d'un fichier ou d'un dossier
--description	-d	La description d'un fichier ou d'un dossier
--title	-i	Le titre d'un fichier ou d'un dossier
--expirationDate	-q	La date d'expiration d'un fichier ou d'un dossier
--expirationStartDate		La date de début d'expiration d'un fichier ou d'un dossier
--expirationEndDate		La date de fin d'expiration d'un fichier ou d'un dossier
--keyword	-k	Le mot-clé d'un fichier ou d'un dossier
--cascade	-x	Indique que les paramètres de sécurité pour un dossier doivent être propagés aux sous-dossiers et aux fichiers
--provider	-f	Le fournisseur de sécurité permettant de récupérer les entités
--createVersion	-b	Indique qu'une nouvelle version d'un fichier doit être créée
--contentLanguage	-g	La langue du contenu d'un fichier ou d'un dossier
--topic		Les rubriques attribuées à un fichier ou un dossier. Vous pouvez indiquer plusieurs valeurs telles que --topic "rubrique1;rubrique2"
--modifiedBy		L'utilisateur qui a modifié un fichier ou un dossier
--mimeType		Le type mime d'un fichier
--createdBy		L'utilisateur qui a créé un fichier ou un dossier
--submittedHierarchy		Indique s'il faut ou non rechercher le dossier <i>Tâches soumises</i>
--propertyName		Le nom d'une propriété personnalisée
--customProperty		La paire valeur/nom d'une propriété personnalisée à mettre à jour
--propertyName		Le nom de la propriété personnalisée pour laquelle récupérer les valeurs valides

- `<expirationStartDate>` est la date de début de l'expiration du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD` (par exemple, `1997-07-16T19:20:30.45+01:00`), où :
 - YYYY = année à quatre chiffres
 - MM = mois à deux chiffres (01 est Janvier, etc.)
 - DD = jour du mois à deux chiffres (01 jusqu'à 31)
 - hh = heure à deux chiffres (00 jusqu'à 23 , sans notification am/pm)
 - mm = minutes à deux chiffres (00 jusqu'à 59)
 - ss = secondes à deux chiffres (00 jusqu'à 59)
 - s = chiffres représentant une fraction décimale d'une seconde, avec une plage valide de 0 à 999
 - TZD = identificateur de fuseau horaire (Z ou +hh:mm ou -hh:mm)
- `<expirationEndDate>` est la date de fin de l'expiration du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<createdStartDate>` est la date de début de la création du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<createdEndDate>` est la date de fin de la création du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<objectModifiedStartDate>` est la date de début modifiée du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<objectModifiedEndDate>` est la date de fin modifiée du fichier/dossier. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<versionModifiedStartDate>` est la date de début modifiée de la version. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `<versionModifiedEndDate>` est la date de fin modifiée de la version. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD`.
- `--submittedHierarchy` Indique que le fichier/dossier figure dans le dossier Tâches soumises.

Tous les paramètres sont facultatifs.

Exemple

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

```
python CADSTool.py advanceSearch --createdStartDate "2009-12-01 00:00:00"
--createdEndDate "2010-12-15 21:33:40" -z
```

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Une fois l'API terminée avec succès, une liste de tous les fichiers et dossiers correspondant aux critères de la recherche est affichée. Ceci comprend généralement les noms de fichiers avec leurs chemins d'accès complets et leurs versions.

- Erreur lors de la recherche des fichiers et des dossiers
- erreur : cette option n'est pas disponible :<option>

L'opération `applySecurity`

Permet de définir la liste de contrôle d'accès de sécurité (ACL) pour un fichier ou un dossier du référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au IBM® SPSS® Collaboration and Deployment Services Repository du fichier ou du dossier auquel appliquer l'ACL de sécurité. Ceci est un paramètre obligatoire.
- `<principal>` (entité) est l'utilisateur (*admin*, par exemple) à appliquer au fichier ou au dossier spécifié dans le cadre de l'ACL. Ceci est un paramètre obligatoire.
- `<permission>` (autorisation) est le type d'autorisation à appliquer au fichier ou dossier indiqué (lecture, écriture, modification, suppression ou propriétaire, par exemple). Ceci est un paramètre obligatoire.
- `<fournisseur>` est le fournisseur de sécurité à utiliser pour la récupération des informations sur les utilisateurs (entités). Ceci est un paramètre facultatif.
- `--cascade` est utilisé lors de la définition de la sécurité pour un dossier, pour propager les paramètres de sécurité à tous les fichiers et sous-dossiers au sein d'un dossier spécifié. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant applique la sécurité à un dossier°:

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

L'exemple suivant applique la sécurité à un dossier et à tous ses fichiers et sous-dossiers°:

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- autorisation <autorisation> définie correctement pour <source>.
- <source> Aucun fichier ou dossier de ce type n'existe. Veuillez recommencer.

- `<autorisation>` Type d'autorisation non valide. Veuillez recommencer.
- `<source>` Erreur lors de la définition de l'ACL de sécurité.

L'opération cascadeSecurity

Propage les paramètres de sécurité d'un dossier à tous les fichiers et sous-dossiers du dossier.

Syntaxe

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

La valeur de `<source>` correspond au chemin complet du dossier dans ce référentiel. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- ACL de sécurité correctement en cascade pour `<source>`.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la mise en cascade de l'ACL de sécurité.

L'opération copyResource

Permet de copier un fichier ou un dossier vers un autre dossier du référentiel. Une fonctionnalité de changement de nom est fournie pour cette API, avec laquelle le fichier/dossier indiqué peut être renommé lors de sa copie. Les cas décrits au début de [L'opération moveResource sur p. 19](#) s'appliquent également à cette API `copyResource`.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au Référentiel de contenu du fichier/dossier à copier. Ceci est un paramètre obligatoire.
- `<target>` (cible) est le chemin d'accès complet au référentiel où le fichier/dossier doit être copié. Ceci est un paramètre obligatoire.

Exemples

L'exemple suivant permet de copier un fichier :

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



```
"/Projects" -z
```

L'exemple suivant permet de copier et de renommer un fichier :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- <source> correctement copié.
- <source> Aucun fichier ou dossier de ce type n'existe. Veuillez recommencer.
- <target> (cible) Aucun dossier de ce type n'existe. Veuillez recommencer.
- <source> Erreur lors de la copie du fichier ou du dossier.

L'opération createFolder

Permet de créer un dossier à un emplacement indiqué dans le référentiel.

Syntaxe

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

La valeur de <source> correspond au chemin complet du nouveau dossier à créer. Ceci est un paramètre obligatoire. Selon le chemin d'accès indiqué, le dossier est créé, y compris ses sous-dossiers.

Exemple

L'exemple suivant permet de créer le dossier *Drafts* s'il n'existe pas.

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- <source> Dossier correctement créé.
- <source> Aucun dossier de ce type n'existe. Veuillez recommencer.
- <dossier> Le dossier existe déjà. Veuillez recommencer.
- <source> Erreur lors de la création du dossier.

L'opération deleteFile

Permet de supprimer un fichier du référentiel, avec toutes ses versions.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du fichier à supprimer. Ceci est un paramètre obligatoire.
- `--submittedHierarchy` permet de supprimer un fichier du dossier Tâches soumises. Ceci est un paramètre facultatif.

Exemple

L'exemple suivant permet de supprimer un fichier du référentiel, avec toutes ses versions°:

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

L'exemple suivant permet de supprimer un fichier du dossier Tâches soumises, avec toutes ses versions°:

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` correctement supprimé.
- `<source>` Aucun fichier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la suppression du fichier.

L'opération deleteFileVersion

Permet de supprimer une version spécifique d'un fichier du référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du fichier à supprimer. Ceci est un paramètre obligatoire.
- `<version>` est la version spécifique du fichier à supprimer. Ceci est un paramètre facultatif.
- `<label>` (étiquette) est l'étiquette du fichier à supprimer. Ceci est un paramètre facultatif.
- `--submittedHierarchy` permet de supprimer une version spécifique d'un fichier du dossier Tâches soumises. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant permet de supprimer une version spécifique d'un fichier°:

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

L'exemple suivant permet de supprimer un fichier avec une étiquette spécifique :

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

L'exemple suivant permet de supprimer un fichier avec une étiquette spécifique du dossier Tâches soumises :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- <source> correctement supprimé.
- <source> Aucun fichier de ce type n'existe. Veuillez recommencer.
- <source> Erreur lors de la suppression du fichier.

L'opération deleteFolder

deleteFolder permet de supprimer un dossier du référentiel, dont tout son contenu.

Syntaxe

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

où :

- <source> est le chemin d'accès complet au référentiel du dossier à supprimer. Ceci est un paramètre obligatoire.
- --submittedHierarchy permet de supprimer une version spécifique du dossier Tâches soumises. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant permet de supprimer un dossier :

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

L'exemple suivant permet de supprimer un dossier du dossier Tâches soumises :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` correctement supprimé.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la suppression du dossier.

L'opération `downloadFile`

Permet de télécharger une version spécifique d'un fichier depuis le répertoire vers le système de fichiers local.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel ou l'URI de l'objet du dossier contenant le fichier à télécharger. L'URI de l'objet peut être obtenu en affichant les propriétés du dossier dans IBM® SPSS® Collaboration and Deployment Services Deployment Manager. Ceci est un paramètre obligatoire.
- `<version>` est la version du fichier à télécharger. Ceci est un paramètre facultatif.
- `<label>` (étiquette) est l'étiquette du fichier à télécharger. Ceci est un paramètre facultatif.
- `<target>` (cible) est le chemin d'accès complet (dans le système de fichiers local) vers lequel le fichier doit être téléchargé.

Exemples

L'exemple suivant permet de télécharger la dernière version du fichier :

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

L'exemple suivant permet de télécharger une version spécifique du fichier à l'aide d'un marqueur de version°:

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

L'exemple suivant permet de télécharger une version avec étiquette du fichier :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Fichier correctement téléchargé.

- `<source>` Aucun fichier de ce type n'existe. Veuillez recommencer.
- `<target>` (cible) Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors du téléchargement du fichier.

L'opération export

Permet de démarrer une exportation depuis le Référentiel de contenu, permettant ainsi de sélectionner les fichiers et dossiers à exporter, et d'enregistrer le fichier d'exportation **.pes* vers le système de fichiers local.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du dossier à exporter. Ceci est un paramètre obligatoire.
- `<target>` (cible) est le chemin d'accès complet (dans le système de fichiers local) pour le fichier d'exportation **.pes* à créer. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` correctement exporté.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors l'exportation du dossier.

L'opération getAccessControlList

Permet de récupérer la liste de contrôle d'accès (ACL) de sécurité pour un fichier/dossier spécifié dans le Référentiel de contenu.

Syntaxe

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

La valeur de `<source>` est le chemin d'accès complet du fichier/dossier. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Aucun fichier ou dossier de ce type n'existe. Veuillez recommencer.
- Erreur lors de la récupération des détails de sécurité pour `<source>`.

L'opération `getAllVersions`

Permet de récupérer une liste de toutes les versions d'un fichier dans le référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du fichier pour lequel récupérer les versions. Ceci est un paramètre obligatoire.
- `--submittedHierarchy` permet de récupérer les versions du dossier Tâches soumises. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant permet de récupérer toutes les versions d'un fichier spécifié :

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

L'exemple suivant permet de récupérer toutes les versions d'un fichier spécifié du dossier Tâches soumises :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Aucun fichier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la récupération des versions du fichier.
- Une fois le processus terminé, les informations de chaque version de fichier sont affichées, notamment celles relatives à l'étiquette et au marqueur de version.

L'opération `getChildren`

Permet de récupérer la liste de tous les fichiers et dossiers dans un dossier spécifié du référentiel.

Syntaxe

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

La valeur de `<source>` est le chemin d'accès complet du dossier. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Une fois la commande validée, tout le contenu du dossier spécifié est répertorié.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de l'obtention des ressources.

L'opération `getCustomPropertyValue`

Permet de récupérer les valeurs valides acceptées par une propriété personnalisée indiquée.

Syntaxe

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

La valeur de `<propertyName>` est le nom de la propriété personnalisée. Ceci est un paramètre facultatif.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<propertyName>` accepte les valeurs comme `<valides>`
- Erreur lors de la récupération des détails de propriétés pour `<propertyName>`.

L'opération `getMetadata`

Permet de récupérer les attributs de métadonnées d'un fichier ou d'un dossier dans le référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du fichier ou du dossier pour lequel récupérer les métadonnées. Pour les dossiers, les attributs `version/label` sont ignorés. Ceci est un paramètre obligatoire.
- `<version>` est la version du fichier ou du dossier pour lequel récupérer les métadonnées. Ceci est un paramètre facultatif.
- `<label>` (étiquette) est l'étiquette du fichier ou du dossier pour lequel récupérer les métadonnées. Ceci est un paramètre facultatif.
- `--submittedHierarchy` permet de récupérer les métadonnées du dossier Tâches soumises. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant récupère les métadonnées pour un dossier°:

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

L'exemple suivant permet de récupérer les métadonnées d'une version avec étiquette d'un fichier :

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

L'exemple suivant permet de récupérer les métadonnées d'une version avec étiquette d'un fichier dans le dossier Tâches soumises :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Aucun fichier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la récupération des métadonnées du fichier.
- Une fois le processus terminé, toutes les informations de métadonnées pour le fichier ou le dossier spécifié sont affichées, notamment les propriétés de métadonnées personnalisées.

L'opération import

Permet d'importer un fichier d'exportation `*.pes` existant depuis le système de fichiers local vers le référentiel.

Syntaxe

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


où :

- *<source>* est le chemin d'accès complet (dans le système de fichiers local) pour le fichier d'exportation *.pes à importer vers le référentiel. Ceci est un paramètre obligatoire.
- *<target>* (cible) est le chemin d'accès complet au référentiel vers lequel importer le fichier d'exportation *.pes. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- *<source>* correctement importé.
- *<source>* Aucun fichier de ce type n'existe. Veuillez recommencer.
- *<target>* (cible) Aucun dossier de ce type n'existe. Veuillez recommencer.
- *<source>* Erreur lors de l'importation du dossier.

L'opération moveResource

Permet de déplacer un fichier ou un dossier vers un autre dossier du référentiel. Une fonctionnalité de changement de nom est fournie pour cette API, avec laquelle le fichier/dossier indiqué peut être renommé lorsqu'il est déplacé. Les cas suivants décrivent le comportement de la fonctionnalité de changement de nom :

Si la source est */Temp Folder/Temp.txt* et la cible est */Demo Folder* :

- **Cas 1:** Si le dossier *Demo Folder* existe, *Temp.txt* est déplacé vers *Demo Folder*.
- **Cas 2:** Si le dossier *Demo Folder* n'existe pas, *Temp.txt* est déplacé vers "/" et renommé en *Demo Folder*.

Si la source est */Temp Folder/Temp.txt* et la cible est */Demo Folder/Abc.dat* :

- **Cas 1:** Si le dossier *Demo Folder* existe, *Temp.txt* est déplacé vers *Demo Folder* et renommé en *Abc.dat*.
- **Cas 2:** Si le dossier *Demo Folder* n'existe pas, un message d'erreur est affiché.

Syntaxe

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

où :

- *<source>* est le chemin d'accès complet au référentiel du fichier/dossier à déplacer. Ceci est un paramètre obligatoire.
- *<target>* (cible) est le chemin d'accès complet au référentiel vers lequel le fichier/dossier doit être déplacé. Ceci est un paramètre obligatoire.

Exemples

L'exemple suivant permet de déplacer un fichier :

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

L'exemple suivant permet de déplacer un dossier :

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

L'exemple suivant permet de déplacer et de renommer un fichier :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` correctement déplacé.
- `<source>` Aucun fichier ou dossier de ce type n'existe. Veuillez recommencer.
- `<target>` (cible) Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors du déplacement du fichier ou du dossier.

L'opération `removeLabel`

Permet de supprimer une étiquette d'un fichier dans le référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet du fichier dans le référentiel. Ceci est un paramètre obligatoire.
- `<label>` (étiquette) est le nom de l'étiquette à supprimer du fichier spécifié. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Etiquette correctement supprimée pour `<source>`.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.

- `<source>` Erreur lors de la suppression de l'étiquette.
- `<étiquette>` Aucune étiquette de ce type n'existe. Veuillez recommencer.

L'opération `removeSecurity`

Permet de supprimer la liste de contrôle d'accès de sécurité (ACL) d'un fichier ou d'un dossier spécifié du référentiel.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet du fichier/dossier duquel supprimer la sécurité. Ceci est un paramètre obligatoire.
- `<principal>` (entité) est l'utilisateur/l'entité (*admin*, par exemple) de laquelle supprimer la sécurité pour le fichier/dossier spécifié. Ceci est un paramètre obligatoire.
- `<fournisseur>` est le fournisseur de sécurité à utiliser pour la récupération des informations sur les utilisateurs (entités). Ceci est un paramètre facultatif.
- `--cascade` est utilisé lors de la suppression de la sécurité d'un dossier, pour supprimer les paramètres de sécurité de tous les fichiers et sous-dossiers d'un dossier spécifié. Ceci est un paramètre facultatif.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Toutes les ACL de sécurité ont été correctement supprimées.
- `<source>` Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la suppression de l'ACL de sécurité.

L'opération `search`

Permet de rechercher des fichiers et des dossiers dans le référentiel. Il en résulte une liste de fichiers/dossiers correspondant aux critères de la recherche, et leurs versions.

Syntaxe

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

La valeur de *<criteria>* est la chaîne de recherche permettant de rechercher les métadonnées pour tous les fichiers et dossiers dans le référentiel. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Une fois la recherche effectuée, une liste de tous les fichiers et dossiers correspondant aux critères de la recherche est affichée. Ceci comprend généralement les noms de fichiers avec leurs chemins d'accès complets et leurs versions.
- *<critère>* Aucun fichier ou dossier ne correspond au critère de la recherche.
- Erreur lors de la recherche des fichiers et des dossiers.

L'opération setLabel

Permet d'appliquer une étiquette à une version d'un fichier dans le référentiel. Si le fichier comporte déjà une étiquette, celle d'origine est supprimée et remplacée par la nouvelle.

Syntaxe

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

où :

- *<source>* est le chemin d'accès complet du fichier dans le référentiel. Ceci est un paramètre obligatoire.
- *<version>* est la version du fichier à laquelle appliquer l'étiquette. Ceci est un paramètre obligatoire.
- *<label>* (étiquette) est le nom de l'étiquette à appliquer à la version spécifiée du fichier. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Etiquette correctement définie pour *<source>*.
- *<source>* Aucun dossier de ce type n'existe. Veuillez recommencer.
- *<source>* Erreur lors de la définition de l'étiquette.

L'opération `setMetadata`

Permet d'appliquer les propriétés de métadonnées aux fichiers et aux dossiers dans le référentiel. [Table 2-3](#) Indique les propriétés de métadonnées et si elles peuvent être appliquées aux fichiers et/ou aux dossiers.

Table 2-3
Propriétés de métadonnées et types de ressources

Propriété de métadonnées	Type de ressource
Auteur	Fichier
Description	Fichier ou dossier
Titre	Fichier ou dossier
Date d'expiration	Fichier ou dossier
Mot clé	Fichier
Rubriques	Fichier
Métadonnées personnalisées	Fichier ou dossier

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet au référentiel du fichier ou du dossier pour lequel définir les métadonnées. Ceci est un paramètre obligatoire.
- `<author>` (auteur) est l'auteur du fichier ou du dossier. Ceci est un paramètre facultatif.
- `<title>` (titre) est le titre du fichier ou du dossier. Ceci est un paramètre facultatif.
- `<description>` est la description du fichier/dossier. Ceci est un paramètre facultatif.
- `<expirationDate>` est la date d'expiration du fichier ou du dossier. Ceci est un paramètre facultatif. Le format de date est `YYYY-MM-DDThh:mm:ss.sTZD` (par exemple, `1997-07-16T19:20:30.45+01:00`), où :
 - YYYY = année à quatre chiffres
 - MM = mois à deux chiffres (01 est Janvier, etc.)
 - DD = jour du mois à deux chiffres (01 jusqu'à 31)
 - hh = heure à deux chiffres (00 jusqu'à 23, sans notification am/pm)
 - mm = minutes à deux chiffres (00 jusqu'à 59)
 - ss = secondes à deux chiffres (00 jusqu'à 59)
 - s = chiffres représentant une fraction décimale d'une seconde, avec une plage valide de 0 à 999
 - TZD = identificateur de fuseau horaire (Z ou +hh:mm ou -hh:mm)
- `<keyword>` (mot clé) est le mot clé du fichier ou du dossier. Ceci est un paramètre facultatif.
- `<version>` est la version spécifique du fichier ou du dossier auquel appliquer les métadonnées. Ceci est un paramètre facultatif.
- `<label>` (étiquette) est la version avec étiquette du fichier/dossier auquel appliquer les métadonnées. Ceci est un paramètre facultatif.

- `<topic>` (rubrique) est la rubrique à appliquer au fichier ou au dossier. Ceci est un paramètre facultatif.
- `<customProperty>` représente les valeurs de propriété personnalisées à appliquer au fichier ou au dossier. Ceci est un paramètre facultatif. Les valeurs sont indiquées sous la forme `<customProperty>=<value>`. Pour appliquer plusieurs propriétés personnalisées, utilisez un point-virgule (;) comme séparateur (`<customProperty>=<value>;<customProperty>=<value>`). Séparez les valeurs de propriétés de sélection multiple avec l'opérateur | (`<customProperty>=opt1|opt2;<customProperty>=value`).

Remarque : Au minimum, un paramètre facultatif doit être fourni pour utiliser l'API `setMetadata`.

Exemple

```
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
```

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Métadonnées correctement définies.
- `<source>` Aucun fichier ou dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de la définition des métadonnées.

L'opération `uploadFile`

`uploadFile` Permet d'enregistrer un fichier dans le Référentiel de contenu depuis le système de fichiers local, avec l'option de création d'une nouvelle version de fichier s'il existe déjà.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet (dans le système de fichiers local) du fichier à transférer. Ceci est un paramètre obligatoire.
- `<target>` (cible) est le chemin d'accès complet du dossier dans le référentiel vers lequel le fichier doit être transféré. Ceci est un paramètre obligatoire.
- `--createVersion` Indique que le fichier spécifié existe déjà et qu'une nouvelle version doit être créée. Ceci est un paramètre facultatif.

Exemples

Dans l'exemple suivant, la cible est un chemin d'accès complet pour le dossier *Drafts* :

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

Si *MyReport.rptdesign* existe déjà dans le dossier */Demo/Drafts*, utilisez le paramètre `--createVersion` :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Fichier correctement téléchargé.
- `<source>` Aucun fichier de ce type n'existe. Veuillez recommencer.
- `<target>` (cible) Aucun dossier de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors du téléchargement du fichier.

Fonctions de gestion des processus

Les scripts de gestion des processus permettent l'utilisation des tâches. Cette zone comprend les fonctionnalités suivantes :

- Exécution des tâches
- Extraction de l'historique des tâches
- Extraction des détails de la tâche

Cette section décrit l'utilisation de scripts de ligne de commande Python pour les fonctions de gestion des processus. Chaque API contient des informations détaillées sur la syntaxe, un exemple et les messages attendus.

Mots clés

[Table 2-4](#) répertorie les mots clés pris en charge pour les API de gestion des processus. La deuxième colonne décrit toute version abrégée facultative des mots clés fournis. Le tableau répertorie uniquement les mots clés spécifiques aux API de gestion des processus. Pour d'autres mots clés qui s'appliquent à la fois aux API de gestion des processus et aux API du référentiel, consultez [Table 2-1](#) and [Table 2-2](#).

Table 2-4
Mots clés des API de gestion des processus

Mot clé	Version courte facultative	Utilisation
--source	-s	La tâche source, dont le chemin d'accès
--target	-t	Le chemin d'accès au dossier cible
--notification	-j	Indique que la tâche sera exécutée avec des notifications
--async	-m	Indique que la tâche sera exécutée de façon asynchrone
--execId	-y	L'ID d'exécution de la tâche
--jobStepName	-q	Le nom de l'étape de la tâche
--log		Indique que les journaux ne doivent pas être supprimés. En cas d'utilisation conjointe avec --target, les journaux sont stockés dans un emplacement spécifié par --target. Sinon, les journaux sont affichés en ligne.

Opérations

Les sections suivantes répertorient toutes les API de scripts de gestion des processus prises en charge pour IBM® SPSS® Collaboration and Deployment Services. La syntaxe et les exemples illustrés contiennent le nombre minimum de paramètres obligatoires.

L'opération deleteJobExecutions

Supprime les objets d'exécution de tâche spécifiés.

Syntaxe

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

La valeur de <execIDs> est une liste d'identificateurs délimités par des espaces pour les exécutions à supprimer. Ceci est un paramètre obligatoire.

Exemples

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- ID d'exécution non spécifié.

L'opération executeJob

Exécute une tâche de façon synchrone ou asynchrone selon les paramètres passés. Dans le cas d'une exécution synchrone, l'API n'effectue aucun renvoi avant la fin de la tâche. Pour une exécution asynchrone, l'API effectue un renvoi après le début de la tâche.

Syntaxe

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

où :

- `<source>` est le chemin d'accès complet de la tâche dans le référentiel. Ceci est un paramètre obligatoire.
- `--notification` est utilisé pour exécuter la tâche avec des notifications. Ceci est un paramètre facultatif.
- `--async` est utilisé pour exécuter la tâche de façon asynchrone. Ceci est un paramètre facultatif.

Exemples

L'exemple suivant permet d'exécuter la tâche de façon synchrone sans notifications :

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

L'exemple suivant permet d'exécuter la tâche de façon synchrone avec des notifications :

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

L'exemple suivant permet d'exécuter la tâche de façon asynchrone sans notifications :

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

L'exemple suivant permet d'exécuter la tâche de façon asynchrone avec des notifications :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- `<source>` Tâche correctement exécutée. L'ID d'exécution de la tâche est `<execId>`.
- `<source>` Aucune tâche de ce type n'existe. Veuillez recommencer.
- `<source>` Erreur lors de l'exécution de la tâche.

L'opération `getJobExecutionDetails`

Répertorie les détails d'exécution d'une tâche spécifique, y compris les étapes et itérations de la tâche.

Syntaxe

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

où :

- `<execId>` est l'identificateur d'exécution de la tâche. Ceci est un paramètre obligatoire.

- `--log` indique que le journal des tâches doit être affiché en ligne. Si le paramètre `--log` n'est pas inclus, tout journal généré par une exécution d'étape de tâche n'est pas affiché. Ceci est un paramètre facultatif.
- `<cible>` est l'emplacement de stockage (dans le système de fichiers local) des journaux. Ceci est un paramètre facultatif, et est uniquement utilisé conjointement avec le paramètre `--log`.

Exemples

L'exemple suivant décrit les détails d'une exécution de tâche spécifique :

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

L'exemple suivant décrit les détails d'une exécution de tâche spécifique, avec le journal affiché en ligne :

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

L'exemple suivant décrit les détails d'une exécution de tâche spécifique, avec les journaux de tâches stockés dans un emplacement spécifique :

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Pour une exécution réussie, tous les détails de cette exécution sont répertoriés pour la tâche, les étapes, et les itérations de la tâche. Les journaux sont affichés en ligne ou enregistrés dans un emplacement spécifié du système de fichiers local.
- `<execId>` Aucune exécution de ce type n'existe. Veuillez recommencer.
- `<execId>` Erreur lors de l'affichage des détails d'une exécution de tâche.
- `--target` ne peut pas être utilisé sans le paramètre `--log`

L'opération `getJobExecutionList`

Répertorie les exécutions en cours et terminées pour une tâche spécifique, pour toutes les versions de la tâche.

Syntaxe

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

La valeur de `<source>` correspond au chemin complet de la tâche dans le référentiel. Ceci est un paramètre obligatoire.

Exemple

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

Messages

Il est possible que les messages suivants s'affichent lors de l'utilisation de cette API :

- Pour une exécution réussie de la tâche spécifiée, tous les détails d'exécution tels que l'ID d'exécution, le nom de la tâche, l'état d'exécution de la tâche, et l'heure de début et de fin de l'exécution de la tâche sont répertoriés.
- <source> Aucune tâche de ce type n'existe. Veuillez recommencer.
- <source> Erreur lors de l'affichage de la liste d'exécutions pour une tâche.

Le module *PESImpl*

L'utilitaire de IBM® SPSS® Collaboration and Deployment Services - Essentials for Python permet l'interaction avec les objets du IBM® SPSS® Collaboration and Deployment Services Repository directement dans les scripts Python. Au sein de votre code Python, importez la classe `PESImpl` depuis le module `pes.api.PESImpl`. Créez un objet `PESImpl` à l'aide des informations de connexion du référentiel auquel se connecter.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl(utilisateur, mot de passe, hôte, port, ssl=True)
```

Les paramètres pour le constructeur `PESImpl` sont les suivants :

- *user* (utilisateur) correspond au nom d'utilisateur
- *password* (mot de passe) correspond au mot de passe associé à l'utilisateur spécifié
- *host* (hôte) désigne le nom du serveur de référentiel
- *port* spécifie le numéro de port pour le serveur de référentiel
- *ssl=True* (vrai) indique que le serveur de référentiel utilise le protocole secure sockets layer (SSL) pour chiffrer les communications. Si le paramètre *ssl* est configuré sur *False* (faux), ou s'il est omis lors de la création de l'objet `PESImpl`, les communications du serveur n'utilisent pas le protocole SSL. Si *ssl=True* (vrai) est utilisé, le serveur de référentiel doit être configuré pour le protocole SSL. Pour plus d'informations, reportez-vous à la documentation de l'administrateur.

Vous pouvez ensuite accéder à des méthodes spécifiques à l'aide de l'objet `pesImpl`.

API de Référentiel de contenu

La génération de scripts du référentiel de contenu permet d'utiliser les ressources de référentiel, telles que les fichiers et les dossiers. Cette zone comprend les fonctionnalités suivantes :

- Création et suppression de dossiers
- Téléchargement et transfert de fichiers
- Exporter et importer des dossiers
- Gestion des étiquettes, de la sécurité et des métadonnées.

Cette section décrit l'API `PESImpl` utilisée pour la manipulation des ressources stockées dans le référentiel. Chaque méthode contient des informations détaillées sur la syntaxe, un exemple et les messages attendus.

Méthodes

Les sections suivantes répertorient toutes les méthodes de référentiel de contenu prises en charge pour IBM® SPSS® Collaboration and Deployment Services.

Remarques :

- Pour toutes les méthodes avec les paramètres facultatifs `Label` et `Version`, utilisez soit `Label`, soit `Version`, mais pas les deux. Si aucun paramètre `Version` ou `Label` n'est indiqué pour un fichier ou un dossier donné, la dernière version est utilisée.
- Pour toutes les méthodes décrites dans ce guide qui nécessitent un chemin d'accès aux fichiers ou aux dossiers dans le référentiel, vous pouvez utiliser soit le chemin d'accès, soit l'URI de l'objet. L'URI de l'objet peut être obtenu en affichant les propriétés de l'objet dans IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- Pour les méthodes nécessitant l'entrée d'un référentiel source ou cible ou de chemins de systèmes de fichiers qui contiennent des caractères Unicode non latins, les chaînes doivent être spécifiées en tant qu'objets Unicode, par exemple :

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

La méthode *advanceSearch*

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

Permet de rechercher les fichiers et les dossiers dans le référentiel selon les divers paramètres transmis comme entrée. Vous pouvez effectuer une recherche pour les éléments suivants :

- Author
- Description
- Title
- Créé par
- Modifié par
- Date de début de l'expiration
- Date de fin de l'expiration
- Type MIME
- Label
- Keyword
- Rubriques
- Date de démarrage créée
- Date de fin créée
- Date de démarrage modifiée de la version
- Date de fin modifiée de l'objet
- Date de démarrage modifiée de l'objet
- Date de démarrage modifiée de la version
- Date de fin modifiée de la version
- URI de dossier parent
- URI de ressource

Remarque : les paramètres `expirationStartDate` et `expirationEndDate` ne fonctionnent pas actuellement lorsqu'ils sont utilisés avec d'autres champs de recherche (tels que le titre, l'auteur, etc).

Table 3-1
Paramètres d'entrée pour `advanceSearch`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>criteriaDict</code>	Requis	Dictionnaire	Le dictionnaire contient la valeur/clé de paire pour laquelle la recherche sera effectuée. Les valeurs de clés acceptables sont les suivantes : <ul style="list-style-type: none"> • <code>author</code> • <code>title</code> • <code>description</code> • <code>createdBy</code> • <code>modifiedBy</code> • <code>expirationStartDate</code> • <code>expirationEndDate</code> • <code>mimeType</code> • <code>étiquette</code> • <code>keyword</code> • <code>topic</code> • <code>createdStartDate</code> • <code>createdEndDate</code> • <code>objectModifiedStartDate</code> • <code>objectModifiedEndDate</code> • <code>versionModifiedStartDate</code> • <code>versionModifiedEndDate</code> • <code>parentURI</code> • <code>uri</code> 	<pre>{ "author": "admin", "title": "search", "label": "label 1" }</pre>
<code>submittedHierarchy</code>	Facultatif	Boolean	Indique s'il faut ou non rechercher le dossier <i>Tâches soumises</i>	True ou False

Table 3-2
Valeur de renvoi pour `advanceSearch`

Type	Description
<code>PageResult</code>	La structure de chaque ligne correspond à une correspondance de la recherche. Pour plus d'informations, reportez-vous à la section La classe PageResult sur p. 55.

Table 3-3
Exceptions pour `advanceSearch`

Type	Description
<code>InsufficientParameterException</code>	Les paramètres obligatoires ne sont pas indiqués.

Exemple : recherche par étiquette et mot clé

L'exemple suivant renvoie toutes les versions de fichiers annotées *Production* qui ont une valeur de mot-clé *Quarterly*.

```
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()
```

Exemple : Recherche par URI

L'échantillon suivant renvoie toutes les versions de fichier du fichier avec l'URI spécifié.

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

La méthode applySecurity

```
applySecurity(source, entité, autorisation, provider, cascade)
```

Permet de définir la liste de contrôle d'accès de sécurité (ACL) pour un fichier ou un dossier du référentiel.

Table 3-4
Paramètres d'entrée d'applySecurity

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>entité</i>	Requis	String	L'utilisateur (tel que <i>admin</i>) à appliquer au fichier ou au dossier indiqué dans le cadre de l'ACL	admin
<i>autorisation</i>	Requis	String	Le type d'autorisation à appliquer au fichier ou au dossier indiqué	READ, WRITE, DELETE, MODIFY_ACL OU OWNER

Champ	Utilisation	Type	Description	Exemple de valeur
<i>provider</i>	Facultatif	String	Le fournisseur de sécurité à utiliser pour appliquer la sécurité aux utilisateurs (<i>Natif</i> , par exemple)	Natif
<i>cascade</i>	Facultatif	Boolean	Propage les paramètres de sécurité à tous les fichiers et sous-dossiers du dossier indiqué	True ou False

Table 3-5

Valeur de renvoi pour `applySecurity`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-6

Exceptions pour `applySecurity`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>IllegalParameterException</code>	Le nom de l'utilisateur ou du fournisseur de sécurité indiqué est incorrect.

Exemple

L'exemple suivant affecte l'autorisation `READ` pour le fichier indiqué à l'utilisateur `icrod`.

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

La méthode `cascadeSecurity`

```
cascadeSecurity(source)
```

Propage les paramètres de sécurité d'un dossier à tous les fichiers et sous-dossiers du dossier.

Table 3-7

Paramètres d'entrée de `cascadeSecurity`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du dossier dans le référentiel	"/Temp Folder" ou "0a58c3670016a7860000 010dcee0eaa28219"

Table 3-8

Valeur de renvoi pour `cascadeSecurity`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-9
Exceptions pour `cascadeSecurity`

Type	Description
<code>ResourceNotFoundException</code>	Le dossier source n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant répercute la sécurité du dossier *Projets* à tous les enfants du dossier.

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

La méthode `copyResource`

`copyResource(source, target)`

Permet de copier un fichier ou un dossier vers un autre dossier du référentiel. Le fichier ou le dossier source indiqué peut être renommé lorsqu'il est copié. Pour plus d'informations sur le changement de nom, reportez-vous à [La méthode `moveResource` sur p. 46](#).

Table 3-10
Paramètres d'entrée de `copyResource`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>target</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du dossier dans lequel copier le fichier. Un nouveau nom de fichier peut être également fourni pour renommer le fichier ou le dossier indiqué lorsqu'il est copié.	"/New Folder" ou "/New Folder/abc.dat"

Table 3-11
Valeur de renvoi pour `copyResource`

Type	Description
String	URI du fichier ou du dossier copié

Table 3-12
Exceptions pour `copyResource`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source ou le dossier cible n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant en copie le dossier *Drafts* dans un dossier nommé *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

```

La méthode createFolder

```
createFolder(source)
```

Permet de créer un dossier à un emplacement indiqué dans le référentiel.

Table 3-13

Paramètres d'entrée de createFolder

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Les dossiers à créer dans le référentiel	"/Temp Folder/Sample Folder"

Table 3-14

Valeur de renvoi pour createFolder

Type	Description
String	URI du dossier créé

Table 3-15

Exceptions pour createFolder

Type	Description
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.
ResourceAlreadyExistsException	Le dossier spécifié existe déjà dans le référentiel.

Exemple

L'exemple suivant crée un dossier nommé *Drafts* en tant qu'enfant du dossier *Demo*. Si un problème survient lors de la création du dossier, un message d'exception est envoyé à la console.

```

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

```

La méthode deleteFile

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

Permet de supprimer un fichier du référentiel. Toutes les versions du fichier sont supprimées.

Table 3-16
Paramètres d'entrée de `deleteFile`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>sub- mitted- Hierar- chy</i>	Facultatif	Boolean	Indique si le fichier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-17
Valeur de renvoi pour `deleteFile`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-18
Exceptions pour `deleteFile`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>IllegalParameterException</code>	La ressource spécifiée à supprimer est un dossier.

Exemple

L'exemple suivant supprime le fichier *MyReport.rptdesign* du référentiel.

```
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."
```

La méthode `deleteFileVersion`

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

Permet de supprimer une version spécifique d'un fichier du référentiel.

Table 3-19
Paramètres d'entrée de `deleteFileVersion`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a78600 00010dcee0eaa28219"
<i>version</i>	Facultatif. Cependant, <i>version</i> ou <i>étiquette</i> doit être spécifié.	String	La version spécifique du fichier à supprimer	"0:2006-08-25 21:15:49.453"
<i>étiquette</i>	Facultatif. Cependant, <i>version</i> ou <i>étiquette</i> doit être spécifié.	String	La version spécifique avec étiquette du fichier à supprimer	"Version 1"
<i>submittedHierarchy</i>	Facultatif	Boolean	Indique si le fichier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-20
Valeur de renvoi pour `deleteFileVersion`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-21
Exceptions pour `deleteFileVersion`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source ou le dossier cible n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>IllegalParameterException</code>	La ressource spécifiée à supprimer est un dossier.

Exemple

L'exemple suivant supprime la version du fichier *MyReport.rptdesign* annoté *Test* du référentiel.

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

La méthode `deleteFolder`

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

Permet de supprimer un dossier et tout son contenu du référentiel

Table 3-22
Paramètres d'entrée de deleteFolder

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du dossier dans le référentiel	"/Temp Folder" ou "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Facultatif	Boolean	Indique si le dossier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-23
Valeur de renvoi pour deleteFolder

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-24
Exceptions pour deleteFolder

Type	Description
ResourceNotFoundException	Le dossier spécifié n'existe pas.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.
IllegalParameterException	La ressource spécifiée à supprimer n'est pas un dossier.

Exemple

L'exemple suivant permet de supprimer le dossier nommé *Drafts* du référentiel. Si un problème survient lors de la suppression du dossier, un message d'exception est envoyé à la console.

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

La méthode downloadFile

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

Permet de télécharger une version spécifique d'un fichier depuis le répertoire vers le système de fichiers local.

Table 3-25
Paramètres d'entrée de `downloadFile`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet au référentiel ou l'URI d'objet du fichier à télécharger	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>target</i>	Requis	String	Le chemin d'accès complet (dans le système de fichiers local) du dossier vers lequel télécharger le fichier	"C:\Temp"
<i>version</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	La version spécifique du fichier à télécharger	"0:2006-08-25 21:15:49.453"
<i>étiquette</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	La version spécifique avec étiquette du fichier à télécharger	"Version 2"

Table 3-26
Valeur de renvoi pour `downloadFile`

Type	Description
Ressource	Conteneur d'informations concernant un objet du référentiel. Pour plus d'informations, reportez-vous à la section La classe Resource sur p. 54.

Table 3-27
Exceptions pour `downloadFile`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source ou le dossier cible n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant télécharge une version annotée *Production* du fichier *MyReport.rptdesign* dans le répertoire *Partagé* sur le système de fichiers local.

```
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")
```

La méthode `exportResource`

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

Permet d'exporter le dossier du référentiel spécifié dans un fichier d'exportation *.pes indiqué sur le système de fichiers local.

Table 3-28
Paramètres d'entrée d'exportResource

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet au référentiel ou l'URI d'objet du dossier à exporter	"/Temp Folder" ou "0a58c3670016a786000010dcee0eaa2 8219"
<i>target</i>	Requis	String	Le chemin d'accès complet (dans le système de fichiers local) et le nom de fichier vers lequel exporter le dossier	"C:\Temp\backup.pes"

Table 3-29
Valeur de renvoi pour exportResource

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-30
Exceptions pour exportResource

Type	Description
ResourceNotFoundException	Le fichier source ou le dossier cible n'existe pas.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.
IllegalParameterException	La cible spécifiée est un dossier. La cible doit être un fichier *.pes.

Exemple

L'exemple suivant exporte le contenu du dossier *Drafts* dans un fichier d'exportation du dossier *backups* du système de fichiers local.

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

La méthode getAccessControlList

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

Permet de récupérer la liste de contrôle d'accès de sécurité (ACL) pour un fichier ou un dossier spécifié du référentiel.

Table 3-31
Paramètres d'entrée de getAccessControlList

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Facultatif	Boolean	Indique si le fichier ou le dossier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-32
Valeur de renvoi pour `getAccessControlList`

Type	Description
Dictionnaire	Un dictionnaire est affiché contenant les noms d'utilisateur et l'autorisation associée. Par exemple : {"admin":"MODIFY_ACL", "Joe":"DELETE"}

Table 3-33
Exceptions pour `getAccessControlList`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source ou le dossier cible n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant imprime l'ACL pour le fichier `MyReport.rptdesign`.

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

La méthode `getAllVersions`

`getAllVersions(source, submittedHierarchy)`

Permet de récupérer une liste de toutes les versions d'un fichier dans le référentiel.

Table 3-34
Paramètres d'entrée de `getAllVersions`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a78600 00010dcee0eaa28219"
<code>submittedHierarchy</code>	Facultatif	Boolean	Indique si le fichier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-35
Valeur de renvoi pour `getAllVersions`

Type	Description
Liste	Une liste des objets de ressource. Consultez La classe Resource sur p. 54 .

Table 3-36
Exceptions pour `getAllVersions`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>IllegalParameterException</code>	La source spécifiée est un dossier.

Exemple

Cet exemple permet de récupérer des informations relatives à toutes les versions du fichier *MyReport.rptdesign*, en imprimant l'auteur, le marqueur de version et les étiquettes de version pour chacune d'elles.

```
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()
```

La méthode getChildren

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

Permet de récupérer une liste de tous les fichiers et dossiers à l'intérieur d'un dossier spécifié du référentiel

Table 3-37

Paramètres d'entrée de getChildren

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du dossier dans le référentiel	"/Temp Folder" ou "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Facultatif	Booléen	Indique si le dossier figure dans le dossier Tâches soumises	True ou False

Table 3-38

Valeur de renvoi pour getChildren

Type	Description
Liste	Une liste des objets de ressource. Consultez La classe Resource sur p. 54 .

Table 3-39

Exceptions pour getChildren

Type	Description
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.
ResourceNotFoundException	Le dossier n'existe pas.

Exemple

L'exemple suivant permet de récupérer les enfants du dossier */Demo/Drafts*, en imprimant le titre, l'auteur et l'identificateur de ressources pour chacun d'eux.

```
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()
```

La méthode `getCustomPropertyValue`

```
getCustomPropertyValue(propertyName)
```

Permet de récupérer les valeurs valides acceptées par une propriété personnalisée indiquée.

Table 3-40
Paramètres d'entrée de `getCustomPropertyValue`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>propertyName</code>	Requis	String	Le nom de la propriété personnalisée	"FreeForm"

Table 3-41
Valeur de renvoi pour `getCustomPropertyValue`

Type	Description
Liste	Permet de renvoyer une liste des valeurs valides que la propriété personnalisée accepte. Si la propriété nécessite une sélection (par exemple, sélection unique ou multisélection), la liste contient toutes les valeurs valides pour la sélection. Si c'est une propriété de forme libre, la liste contient le type de données que la propriété accepte (par exemple, <code>String</code> , <code>Date</code> ou <code>Number</code>).

Table 3-42
Exceptions pour `getCustomPropertyValue`

Type	Description
<code>ResourceNotFoundException</code>	La propriété spécifiée n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant a accès aux valeurs de la propriété personnalisée `Language`.

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

La méthode `getMetadata`

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

Permet de récupérer les attributs des métadonnées d'un fichier ou d'un dossier dans le référentiel, notamment des propriétés personnalisées et des informations sur les rubriques.

Table 3-43
Paramètres d'entrée de *getMetadata*

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a78600 00010dcee0eaa28219"
<i>version</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	La version spécifique du fichier ou du dossier	"0:2006-08-25 21:15:49.453"
<i>étiquette</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	La version spécifique avec étiquette du fichier ou du dossier	"Version 1"
<i>submittedHierarchy</i>	Facultatif	Boolean	Indique si le fichier figure dans le dossier <i>Tâches soumises</i>	True ou False

Table 3-44
Valeur de renvoi pour *getMetadata*

Type	Description
Ressource	Conteneur d'informations concernant un objet du référentiel. Pour plus d'informations, reportez-vous à la section La classe Resource sur p. 54.

Table 3-45
Exceptions pour *getMetadata*

Type	Description
ResourceNotFoundException	Le fichier ou le dossier source n'existe pas.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant permet d'accéder à l'identificateur de ressources pour le dossier */Demo/Drafts*.

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

La méthode *importResource*

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

Permet d'importer un fichier d'exportation *.pes existant depuis le système de fichiers local vers le référentiel.

Table 3-46
Paramètres d'entrée d'importResource

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet (dans le système de fichiers local) du fichier à importer	"C:\Temp\New.pes"
<i>target</i>	Requis	String	Le chemin d'accès complet au référentiel ou l'URI d'objet du dossier dans lequel réaliser l'importation	"/Temp Folder" ou "0a58c3670016a7860000 010dcee0eaa28219"

Table 3-47
Valeur de renvoi pour importResource

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-48
Exceptions pour importResource

Type	Description
ResourceNotFoundException	Le fichier source ou le dossier cible n'existe pas.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant .

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

La méthode moveResource

`moveResource(source, target)`

Permet de déplacer un fichier ou un dossier vers un autre dossier du référentiel. Un fichier source spécifié peut-être renommé lorsqu'il est déplacé, le type et l'existence de la cible déterminant le nom final. Le tableau suivant décrit le comportement de la fonctionnalité de changement de nom lors du déplacement d'un fichier :

Table 3-49
Changement de nom d'un fichier

Type de cible	Le dossier cible existe	Le dossier cible n'existe pas
dossier	Le fichier source devient un enfant du dossier cible.	Le fichier source est déplacé dans un dossier parent de la cible spécifiée et est renommé avec le nom du dossier cible.
fichier	Le fichier source est déplacé dans le dossier qui contient le fichier cible et est renommé avec le nom de la cible.	Erreur trouvée.

Par exemple, si la source est le fichier */Temp Folder/Temp.txt* et que la cible spécifiée est le dossier */Demo Folder*, les résultats suivants peuvent survenir :

- Si le dossier *Demo Folder* existe, *Temp.txt* est déplacé vers *Demo Folder*.
- Si le dossier *Demo Folder* n'existe pas, *Temp.txt* est déplacé vers "/" et renommé en *Demo Folder*.

Ou, si la source est */Temp Folder/Temp.txt* et que la cible spécifiée est le fichier */Demo Folder/Abc.dat*, les résultats suivants peuvent survenir :

- Si le dossier *Demo Folder* existe, *Temp.txt* est déplacé vers *Demo Folder* et renommé en *Abc.dat*.
- Si le dossier *Demo Folder* n'existe pas, un message d'erreur est affiché.

Table 3-50
Paramètres d'entrée de *moveResource*

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	<code>"/Temp Folder/Temp.txt"</code> ou <code>"0a58c3670016a786000010dcee0eaa28219"</code>
<i>target</i>	Requis	String	Le chemin d'accès complet ou l'URI du dossier vers lequel déplacer le fichier. Un nouveau nom de fichier peut être également fourni pour renommer le fichier ou le dossier indiqué lorsqu'il est déplacé.	<code>"/New Folder"</code> ou <code>"/New Folder/abc.dat"</code>

Table 3-51
Valeur de renvoi pour *moveResource*

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-52
Exceptions pour *moveResource*

Type	Description
<code>ResourceNotFoundException</code>	La source spécifiée n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant .

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

La méthode `removeLabel`

```
removeLabel (source, étiquette)
```

Permet de supprimer une étiquette d'un fichier dans le référentiel.

Table 3-53

Paramètres d'entrée de `removeLabel`

Champ	Utilisation	Type	Exemple de valeur	Description
<i>source</i>	Requis	String	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"	Le chemin d'accès complet ou l'URI de l'objet du fichier dans le référentiel
<i>étiquette</i>	Requis	String	"Version 1"	Le nom d'étiquette à supprimer

Table 3-54

Valeur de renvoi pour `removeLabel`

Type	Description
String	URI du fichier mis à jour

Table 3-55

Exceptions pour `removeLabel`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant permet de supprimer l'étiquette *Draft* du fichier *MyReport.rptdesign*.

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

La méthode `removeSecurity`

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

Permet de supprimer la liste de contrôle d'accès de sécurité (ACL) d'un fichier ou d'un dossier spécifié du référentiel.

Table 3-56

Paramètres d'entrée de `removeSecurity`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>entité</i>	Requis	String	L'utilisateur (<i>admin</i> , par exemple) à supprimer du fichier ou du dossier indiqué	admin

Champ	Utilisation	Type	Description	Exemple de valeur
<i>provider</i>	Facultatif	String	Le fournisseur de sécurité (<i>Native</i> , par exemple) à utiliser pour obtenir les informations sur les utilisateurs	Natif
<i>cascade</i>	Facultatif	Boolean	Propage les paramètres de sécurité à tous les fichiers et sous-dossiers du dossier indiqué	True ou False

Table 3-57

Valeur de renvoi pour `removeSecurity`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Table 3-58

Exceptions pour `removeSecurity`

Type	Description
<code>ResourceNotFoundException</code>	Le fichier source ou le dossier cible n'existe pas.
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>IllegalParameterException</code>	Le nom de l'utilisateur ou du fournisseur de sécurité indiqué est incorrect.

Exemple

L'exemple suivant permet de supprimer l'ACL (liste de contrôle d'accès de sécurité) de l'entité *icrod* à partir du fichier *MyReport.rptdesign*.

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

La méthode `search`

```
search(criteria)
```

Permet de rechercher des fichiers dans le référentiel en renvoyant une liste des versions de fichiers dont le contenu des métadonnées correspond aux critères de recherche.

Table 3-59

Paramètres d'entrée de `search`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>criteria</i>	Requis	String	La valeur permettant de rechercher les métadonnées de fichiers	"Age"

Table 3-60

Valeur de renvoi pour `search`

Type	Description
<code>PageResult</code>	La structure de chaque ligne correspond à une correspondance de la recherche. Pour plus d'informations, reportez-vous à la section La classe <code>PageResult</code> sur p. 55.

Table 3-61
Exceptions pour search

Type	Description
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.

Exemple

Les exemples suivants permettent de rechercher les versions des fichiers comportant le texte *Quarterly* dans les champs de métadonnées.

```
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()
```

La méthode setLabel

```
setLabel(source, version, étiquette)
```

Permet d'appliquer une étiquette à une version d'un fichier dans le référentiel. Si le fichier comporte déjà une étiquette, celle d'origine est remplacée par la nouvelle.

Table 3-62
Paramètres d'entrée de setLabel

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>version</i>	Requis	String	La version spécifique du fichier	"0:2006-08-25 21:15:49.453"
<i>étiquette</i>	Requis	String	L'étiquette à appliquer au fichier	"Version 1"

Table 3-63
Valeur de renvoi pour setLabel

Type	Description
String	URI du fichier mis à jour

Table 3-64
Exceptions pour setLabel

Type	Description
ResourceNotFoundException	Le fichier ou la version source n'existe pas.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.

Exemple

L'exemple suivant permet d'affecter l'étiquette *Beta* à la seconde version du fichier *MyReport.rptdesign*. La méthode `getVersionMarker` pour un objet `Resource` renvoie le marqueur de la version que vous souhaitez étiqueter.

```
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")
```

La méthode setMetadata

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

Permet d'appliquer les propriétés de métadonnées aux fichiers et aux dossiers dans le référentiel. Le tableau suivant indique les propriétés de métadonnées et si elles peuvent être appliquées aux fichiers et/ou aux dossiers.

Table 3-65
Propriétés des objets du référentiel

Propriété de métadonnées	Type de ressource
Auteur	Fichier
Description	Fichier ou dossier
Titre	Fichier ou dossier
Date d'expiration	Fichier ou dossier
Mot clé	Fichier
Rubriques	Fichier
Métadonnées personnalisées	Fichier ou dossier

Table 3-66
Paramètres d'entrée de setMetadata

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet ou l'URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000 010dcee0eaa28219"
<i>version</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	La version spécifique du fichier à télécharger	"0:2006-08-25 21:15:49.453"

Champ	Utilisation	Type	Description	Exemple de valeur
<i>étiquette</i>	Facultatif. La version ou l'étiquette peut être indiquée.	String	L'étiquette de la version spécifique	"Label 1"
<i>props</i>	Requis	Dictionnaire	Contient toutes les métadonnées à définir, dans le dictionnaire avec les noms de métadonnées comme clés. Comme illustré dans la colonne Exemple de valeur, elle utilise la liste comme valeur provenant de topic et Dictionary comme customProperty. Pour le reste des métadonnées, il utilise une chaîne.	{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }

Table 3-67
Valeur de renvoi pour `setMetadata`

Type	Description
String	URI du fichier ou du dossier pour lequel les métadonnées ont été définies

Table 3-68
Exceptions pour `setMetadata`

Type	Description
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>ResourceNotFoundException</code>	Le fichier ou le dossier source n'existe pas.

Exemple

L'exemple suivant permet d'affecter le mot-clé *Quarterly* à la version *Production* du fichier *MyReport.rptdesign*.

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

La méthode `uploadFile`

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

Permet d'enregistrer un fichier dans le référentiel depuis le système de fichiers local, avec l'option de création d'une nouvelle version du fichier si celle-ci existe déjà.

Table 3-69
Paramètres d'entrée d'uploadFile

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet (dans le système de fichiers local) du fichier à transférer	"C:\Temp\Temp.txt"
<i>target</i>	Requis	String	Le chemin d'accès complet du dossier de destination dans le référentiel où le fichier doit être transféré	"/Temp Folder"
<i>versionFlag</i>	Facultatif	Boolean	Si le fichier indiqué existe déjà, une nouvelle version du fichier est créée	True ou False

Table 3-70
Valeur de renvoi pour uploadFile

Type	Description
String	URI du fichier transféré

Table 3-71
Exceptions pour uploadFile

Type	Description
ResourceNotFoundException	Le fichier source ou le dossier cible n'existe pas.
ResourceAlreadyExistsException	Un fichier ou un dossier avec le même nom que le fichier source existe dans le dossier cible et le paramètre <i>versionFlag</i> n'est pas spécifié.
InsufficientParameterException	Les paramètres requis ne sont pas indiqués.

Exemple

Cet exemple permet de transférer le fichier *MyReport.rptdesign* dans le dossier */Demo/Drafts* du référentiel. Si le fichier indiqué existe déjà, une nouvelle version du fichier est transférée à l'aide du paramètre *versionFlag*.

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

Classes wrapper

L'API PESImpl comprend des classes servant de wrappers pour des objets renvoyés des services Web appelés par les méthodes de référentiel de contenu. Ces classes wrapper fournissent une interface pour l'affichage des informations renvoyées par les méthodes.

La classe Resource

La classe `Resource` sert de wrapper simplifié au l'objet de référentiel `ResourceSpecifier.Resource`, permettant d'accéder aux informations propres à l'objet. Outre les métadonnées standard associées aux objets du référentiel, cette classe comprend toutes les informations de métadonnées personnalisées définies pour des objets du référentiel.

Table 3-72 répertorie toutes les méthodes disponibles dans la classe `Resource`.

Table 3-72
Méthodes pour la classe `Resource`

Nom de la méthode	Description
<code>getAccessControlList</code>	Permet de renvoyer un dictionnaire des autorisations de sécurité d'un objet. Contient le nom de l'utilisateur comme clé et seule l'autorisation la plus haute accordée à l'utilisateur. Par exemple : Si l'utilisateur <i>Joe</i> possède l'autorisation de <i>suppression</i> pour la ressource <i>X</i> , alors <code>getAccessControlList</code> de l'objet ressource représentant <i>X</i> renvoie <code>{ 'Joe' : 'DELETE' }</code> et pas toutes les trois autorisations (lecture, écriture, suppression) de l'appel du service Web.
<code>getOwner</code>	Renvoie le nom du propriétaire de l'objet sous forme de chaîne
<code>getAuthor</code>	Renvoie le nom de l'auteur de l'objet sous forme de chaîne
<code>getContentSize</code>	Renvoie la taille de l'objet
<code>getCreatedBy</code>	Renvoie le nom de l'utilisateur qui a créé l'objet sous forme de chaîne
<code>getCreationDate</code>	Renvoie la date de création de l'objet sous forme d'objet <code>datetime</code>
<code>getDescription</code>	Renvoie la description de l'objet sous forme de liste
<code>getDescriptionLanguage</code>	Renvoie la langue de l'objet sous forme de liste
<code>getExpirationDate</code>	Renvoie la date d'expiration de l'objet sous forme d'objet <code>datetime</code>
<code>isExpired</code>	Indique si l'objet spécifié a expiré ou non
<code>getMIMEType</code>	Renvoie le type MIME de l'objet sous forme de chaîne
<code>getModificationDate</code>	Renvoie la date de dernière modification de l'objet sous forme d'objet <code>datetime</code>
<code>getObjectCreationDate</code>	Renvoie la date de création de l'objet sous forme d'objet <code>datetime</code>
<code>getObjectLastModifiedBy</code>	Renvoie l'utilisateur qui a modifié pour la dernière fois l'objet sous forme de chaîne
<code>getObjectLastModifiedDate</code>	Renvoie la date de dernière modification de l'objet sous forme d'objet <code>datetime</code>
<code>getResourceID</code>	Renvoie l'identificateur de ressources de l'objet sous forme de chaîne
<code>getResourcePath</code>	Renvoie le chemin d'accès de l'objet spécifié sous forme de chaîne
<code>getTitle</code>	Renvoie le titre de l'objet sous forme de chaîne
<code>getTopicList</code>	Renvoie la liste de rubriques pour l'objet
<code>getVersionMarker</code>	Renvoie la version de l'objet sous forme de chaîne.
<code>getVersionLabel</code>	Renvoie l'étiquette de l'objet sous forme de chaîne
<code>getCustomMetadata</code>	Renvoie des propriétés personnalisées associées à l'objet sous forme de dictionnaire
<code>getKeywordList</code>	Renvoie une liste de mots-clés associés à l'objet

La classe *IdentificationSpecifieur*

Cette classe se comporte comme un wrapper simplifié pour l'objet référentiel `IdentificationSpecifieur`, permettant d'accéder aux données propres à l'identification de l'objet. [Table 3-73](#) répertorie toutes les méthodes disponibles dans la classe `IdentificationSpecifieur`.

Table 3-73
Méthodes pour la classe *IdentificationSpecifieur*

Nom de la méthode	Description
<code>getIdentifieur</code>	Renvoie la valeur de l'identificateur d'un objet sous forme de chaîne
<code>getVersionMarker</code>	Renvoie la version d'un l'objet sous forme de chaîne
<code>getVersionLabel</code>	Renvoie l'étiquette appliquée à un objet sous forme de chaîne

La classe *PageResult*

Cette classe `PageResult` sert de conteneur pour les résultats de la recherche. Une correspondance individuelle dans les résultats correspond à une ligne dans l'objet `PageResult`. Par exemple, une recherche qui renvoie quatre ressources fournit un objet `PageResult` contenant quatre lignes. [Table 3-74](#) répertorie toutes les méthodes disponibles dans la classe `PageResult`.

Table 3-74
Méthodes pour la classe *PageResult*

Nom de la méthode	Description
<code>getRows</code>	Renvoie une liste d'objets <code>SearchRow</code> . Pour plus d'informations, reportez-vous à la section La classe <code>SearchRow</code> sur p. 55.

La classe *SearchRow*

La classe `SearchRow` sert de conteneur pour les informations au niveau objet relatives à un résultat de recherche individuelle. Vous pouvez accéder aux métadonnées relatives à un objet à l'aide des méthodes de cette classe. [Table 3-75](#) répertorie toutes les méthodes disponibles dans la classe `SearchRow`.

Table 3-75
Méthodes pour la classe *SearchRow*

Nom de la méthode	Description
<code>getTitle</code>	Renvoie le nom du fichier ou du dossier
<code>getAuthor</code>	Renvoie l'auteur du fichier ou du dossier
<code>getMimeType</code>	Renvoie le type MIME du fichier
<code>getObjectLastModifiedBy</code>	Renvoie l'utilisateur qui a modifié pour la dernière fois le fichier ou le dossier
<code>getModified</code>	Renvoie la date et heure de la dernière modification du fichier ou du dossier
<code>getFolderPath</code>	Renvoie l'emplacement du fichier ou du dossier

Nom de la méthode	Description
getFolder	Renvoie le nom du dossier parent du fichier ou du dossier
getParentURI	Renvoie l'URI de l'objet du parent
getTopic	Renvoie les rubriques associées au fichier ou au dossier
getChildRow	Renvoie la liste des objets <code>SearchChildRow</code> (pour plus d'informations, consultez la section suivante)

Pour accéder aux informations au niveau version d'un objet, utilisez la méthode `getChildRow` pour renvoyer les lignes enfants correspondants aux versions de l'objet.

La classe `SearchChildRow`

La classe `SearchChildRow` sert de conteneur pour les informations au niveau de la version, relatives à un résultat de recherche individuelle. Vous pouvez accéder aux métadonnées relatives à la version d'un objet à l'aide des méthodes de cette classe. [Table 3-76](#) répertorie toutes les méthodes disponibles dans la classe `SearchChildRow`.

Table 3-76
Méthodes pour la classe `SearchChildRow`

Nom de la méthode	Description
getExpirationDate	Renvoie la date d'expiration du fichier ou du dossier
getKeyword	Renvoie les mots clés associés à la version du fichier ou à la version du dossier
getVersionLabel	Renvoie l'étiquette de version du fichier ou du dossier
getDescription	Renvoie la description du fichier ou du dossier
getLanguage	Renvoie la langue
getVersionCreationDate	Renvoie la date et heure de création du fichier ou du dossier
getVersionMarker	Renvoie le marqueur de version du fichier ou du dossier
getUri	Renvoie l'URI de l'objet du fichier ou du dossier

API de gestion des processus

Les scripts de gestion des processus permettent l'utilisation des tâches. Cette zone comprend les fonctionnalités suivantes :

- Exécution des tâches
- Extraction de l'historique des tâches
- Extraction des détails de la tâche

Cette section décrit les méthodes PESImpl utilisées pour la manipulation des tâches stockées dans le référentiel. Chaque méthode contient des informations détaillées sur la syntaxe, un exemple et les messages attendus.

Méthodes

Les sections suivantes répertorient toutes les méthodes de scripts de gestion des processus prises en charge pour IBM® SPSS® Collaboration and Deployment Services.

Remarque : Pour toutes les méthodes décrites dans ce guide qui nécessitent un chemin d'accès aux fichiers ou aux dossiers dans le référentiel, vous pouvez utiliser soit le chemin d'accès, soit l'URI de l'objet. L'URI de l'objet peut être obtenu en affichant les propriétés de l'objet dans IBM® SPSS® Collaboration and Deployment Services Deployment Manager.

La méthode *cancelJob*

```
cancelJob(executionId)
```

Permet d'annuler une tâche en cours.

Table 3-77

Paramètres d'entrée de *cancelJob*

Champ	Utilisation	Type	Description	Exemple de valeur
<i>executionId</i>	Requis	String	ID d'exécution de la tâche	0a58c33d002ce90800 00010e0ccf7b01800e

Table 3-78

Valeur de renvoi pour *cancelJob*

Type	Description
Boolean	Renvoie un message de succès lors de l'annulation de la tâche

Exemple

Cet exemple présente l'arrêt de l'exécution de la tâche *Reports*.

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

La méthode *deleteJobExecutions*

```
deleteJobExecutions(executionId)
```

Supprime une ou plusieurs exécutions de tâches

Table 3-79
Paramètres d'entrée de `deleteJobExecutions`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>executionId</code>	Requis	Liste	Liste des identificateurs d'exécution de tâche	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

Table 3-80
Valeur de renvoi pour `deleteJobExecutions`

Type	Description
Boolean	True (vrai) ou False (faux) selon l'exécution réussie ou non de la méthode.

Exemple

Cet exemple présente la suppression des exécutions de la tâche *Reports*.

```
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)
```

La méthode `executeJob`

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

Exécute une tâche de façon synchrone ou asynchrone selon les paramètres passés. Dans le cas d'une exécution synchrone, la méthode n'effectue aucun renvoi avant la fin de la tâche. Pour une exécution asynchrone, la méthode effectue un renvoi après le début de la tâche.

Table 3-81
Paramètres d'entrée de la méthode `executeJob`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	String	Le chemin d'accès complet (dans le système de fichiers local) du fichier à transférer	"C:\Temp\Temp.txt"
<code>notification</code>	Facultatif	Boolean	Indique si la tâche est exécutée avec ou sans notifications. La valeur par défaut est <code>False</code> .	True ou False
<code>asynchronous</code>	Facultatif	Boolean	Indique si la tâche est exécutée ou non de façon asynchrone. La valeur par défaut est <code>False</code> .	True ou False

Table 3-82
Valeur de renvoi pour `executeJob`

Type	Description
String	L'identifiant unique de l'exécution de la tâche. Cet identifiant est utilisé comme référence à l'exécution d'une tâche particulière.

Exemple

Cet exemple présente le début de l'exécution de la tâche *Reports* de manière asynchrone avec les notifications.

```
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
```

La méthode `getJobExecutionDetails`

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

Répertorie les détails d'exécution d'une tâche spécifique, notamment les étapes et itérations de la tâche

Table 3-83
Paramètres d'entrée de `getJobExecutionDetails`

Champ	Utilisation	Type	Description	Exemple de valeur
<code>executionId</code>	Requis	String	L'ID d'exécution de la tâche	0a58c33d002ce9080000 010e0ccf7b01800e
<code>log</code>	Facultatif	Boolean	Indique si le journal de tâche est affiché en ligne	True ou False
<code>target</code>	Facultatif	String	L'emplacement de stockage (dans le système de fichiers local) des journaux. Uniquement utilisé conjointement avec le paramètre <code>--log</code> .	"c:\logs"

Table 3-84
Valeur de renvoi pour `getJobExecutionDetails`

Type	Description
<code>jobExecutionDetails</code>	Détails sur l'exécution d'une tâche. Pour plus d'informations, reportez-vous à la section La classe <code>jobExecutionDetails</code> sur p. 61.

Exemple

Cet exemple présente la récupération des informations sur les exécutions des étapes de la tâche dont l'identifiant est `execId`, tout en envoyant les résultats de chaque étape vers la console.

```
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()

```

La méthode `getJobExecutionList`

```
getJobExecutionList(source)
```

Permet de répertorier les exécutions d'une tâche spécifique, dont des tâches en cours et terminées, pour toutes les versions de la tâche.

Table 3-85

Paramètres d'entrée de `getJobExecutionList`

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	String	Le chemin d'accès complet de la tâche dans le référentiel.	"/testJob"

Table 3-86

Valeur de renvoi pour `getJobExecutionList`

Type	Description
PageResult	Conteneur de la liste des exécutions d'une tâche. Consultez Pour plus d'informations, reportez-vous à la section La classe PageResult sur p. 61..

Exemple

Cet exemple présente la récupération des exécutions de la tâche *Reports*, tout en envoyant les informations sur chaque exécution vers la console.

```

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()

```

Classes wrapper

L'API `PESImpl` comprend des classes servant de wrappers pour des objets renvoyés des services Web appelés par les méthodes de gestion des processus. Ces classes wrapper fournissent une interface pour l'affichage des informations renvoyées par les méthodes.

La classe `PageResult`

Cette classe `PageResult` sert de conteneur pour les résultats d'exécution de la tâche, permettant de récupérer les données spécifiques à l'exécution de la tâche. Une exécution de tâche individuelle correspond à une ligne de l'objet `PageResult`. Par exemple, une tâche exécutée quatre fois correspond à un objet `PageResult` contenant quatre lignes. [Table 3-87](#) répertorie toutes les méthodes disponibles dans la classe `PageResult`.

Table 3-87
Méthodes pour la classe `PageResult`

Nom de la méthode	Description
<code>getRows</code>	Renvoie une liste d'objets <code>Row</code> , chacun représentant l'exécution d'une tâche. Pour plus d'informations, reportez-vous à la section La classe <code>Row</code> sur p. 61.

La classe `Row`

La classe `Row` sert de conteneur aux informations de niveau de tâche, relatives à l'exécution d'une tâche. Vous pouvez accéder aux métadonnées relatives à l'exécution d'une tâche à l'aide des méthodes de cette classe. [Table 3-88](#) répertorie toutes les méthodes disponibles dans la classe `Row`.

Table 3-88
Méthodes pour la classe `Row`

Nom de la méthode	Description
<code>getObjId</code>	Renvoie l'ID d'exécution de la tâche
<code>getPath</code>	Renvoie le chemin d'accès de la tâche
<code>getVersionMarker</code>	Renvoie le marqueur de version de la tâche exécutée
<code>getVersionLabel</code>	Renvoie l'étiquette de version de la tâche exécutée
<code>getEventObjId</code>	Renvoie l'ID d'événement de la tâche exécutée
<code>getEventState</code>	Renvoie l'état de la tâche en cours
<code>getEventCompletionCode</code>	Renvoie le code d'achèvement de la tâche
<code>getEventStartDateTime</code>	Renvoie la date et l'heure de début de la tâche
<code>getEventEndDateTime</code>	Renvoie la date et l'heure de fin de la tâche
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de la tâche

La classe `jobExecutionDetails`

Cette classe est renvoyée depuis la méthode `getJobExecutionDetails`. Elle stocke les détails de l'exécution d'une tâche et inclut une liste d'objets `jobStepExecution` fournissant des informations sur chaque étape de la tâche. [Table 3-89](#) répertorie toutes les méthodes disponibles dans la classe `jobExecutionDetails`.

Table 3-89
Méthodes pour la classe `jobExecutionDetails`

Nom de la méthode	Description
<code>getJobStepDetails</code>	Renvoie une liste d'objets <code>jobStepExecutionDetails</code> . Pour plus d'informations, reportez-vous à la section La classe <code>jobStepExecutionDetails</code> sur p. 62.
<code>getArtifactLocation</code>	Renvoie une liste des emplacements d'artefacts de la tâche
<code>getCompletionCode</code>	Renvoie le code d'achèvement de l'exécution de la tâche
<code>getEndDateTime</code>	Renvoie la date et l'heure de fin de l'exécution de la tâche
<code>getEventName</code>	Renvoie le nom d'événement de l'exécution de la tâche
<code>getEventUUID</code>	Renvoie l'ID d'événement de l'exécution de la tâche
<code>getExecutionState</code>	Renvoie l'état de l'exécution de la tâche
<code>getExecutionSuccess</code>	Renvoie l'état de réussite ou d'échec de l'exécution de la tâche
<code>getExecutionWarning</code>	Indique s'il y avait ou non des avertissements
<code>getLog</code>	Renvoie le journal (sous forme de chaîne) généré
<code>getNotificationEnabled</code>	Indique si les notifications par e-mail sont activées ou non
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'exécution de la tâche
<code>getStartDateTime</code>	Renvoie la date et l'heure de début de l'exécution de la tâche
<code>getUserName</code>	Renvoie le nom de l'utilisateur qui a exécuté la tâche
<code>getUUID</code>	Renvoie l'ID d'exécution de la tâche

La classe `jobStepExecutionDetails`

Cette classe stocke les détails d'exécution d'une étape de tâche ainsi qu'une liste des objets `jobStepChildExecutionDetails`. Cette classe contient l'objet `ExecutionDetails`, auquel il délègue tous ses appels de méthodes. [Table 3-90](#) répertorie toutes les méthodes disponibles dans la classe `jobStepExecutionDetails`.

Table 3-90
Méthodes pour la classe `jobStepExecutionDetails`

Nom de la méthode	Description
<code>getJobStepChldExecutionList</code>	Renvoie une liste d'objets <code>jobStepChildExecutionDetails</code> . Pour plus d'informations, reportez-vous à la section La classe <code>jobStepChildExecutionDetails</code> sur p. 63 .
<code>getArtifactLocation</code>	Renvoie une liste des emplacements d'artefacts de l'étape de la tâche
<code>getCompletionCode</code>	Renvoie le code d'achèvement de l'étape de la tâche
<code>getEndDateTime</code>	Renvoie la date et l'heure de fin de l'étape de la tâche
<code>getEventName</code>	Renvoie le nom d'événement de l'étape de la tâche
<code>getEventUUID</code>	Renvoie l'ID d'événement de l'étape de la tâche
<code>getExecutionState</code>	Renvoie l'état d'exécution de l'étape de la tâche
<code>getExecutionSuccess</code>	Renvoie l'état de réussite ou d'échec de l'étape de la tâche
<code>getExecutionWarning</code>	Indique s'il y avait ou non des avertissements
<code>getLog</code>	Renvoie le journal (sous forme de chaîne) généré
<code>getNotificationEnabled</code>	Indique si les notifications par e-mail sont activées ou non
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'étape de la tâche
<code>getStartDateTime</code>	Renvoie la date et l'heure de début de l'étape de la tâche
<code>getUserName</code>	Renvoie le nom de l'utilisateur qui a exécuté l'étape de la tâche
<code>getUUID</code>	Renvoie l'ID d'exécution de l'étape de la tâche

La classe `jobStepChildExecutionDetails`

La classe `jobStepChildExecutionDetails` sert de conteneur aux exécutions enfants des étapes de tâches individuelles. Par exemple, une étape de tâche de rapport itératif produit une exécution enfant pour chaque itération de l'étape. Vous pouvez accéder aux métadonnées relatives aux exécutions enfants à l'aide des méthodes de cette classe. [Table 3-91](#) répertorie toutes les méthodes disponibles dans la classe `jobStepChildExecutionDetails`.

Table 3-91
Méthodes pour la classe *jobStepChildExecutionDetails*

Nom de la méthode	Description
<code>getArtifactLocation</code>	Renvoie une liste des emplacements d'artefacts des exécutions enfants
<code>getCompletionCode</code>	Renvoie le code d'achèvement de l'exécution enfant
<code>getEndDateTime</code>	Renvoie la date et l'heure de fin de l'exécution enfant
<code>getEventName</code>	Renvoie le nom d'événement de l'exécution enfant
<code>getEventUUID</code>	Renvoie l'ID d'événement de l'exécution enfant
<code>getExecutionState</code>	Renvoie l'état de l'exécution enfant
<code>getExecutionSuccess</code>	Renvoie l'état de réussite ou d'échec de l'exécution enfant
<code>getExecutionWarning</code>	Indique s'il y avait ou non des avertissements
<code>getLog</code>	Renvoie le journal (sous forme de chaîne) généré
<code>getNotificationEnabled</code>	Indique si les notifications par e-mail sont activées
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'exécution enfant
<code>getStartDateTime</code>	Renvoie la date et l'heure de début de l'exécution enfant
<code>getUserName</code>	Renvoie le nom de l'utilisateur qui a procédé à l'exécution enfant
<code>getUUID</code>	Renvoie l'ID de l'exécution enfant

Exemples de scripts

Des exemples de scripts illustrant l'utilisation de la classe `PESImpl` sont installés dans le répertoire suivant :

<installation location>\Lib\site-packages\samples

Ces scripts peuvent effectuer un grand nombre de tâches, notamment :

- suppression des éléments expirés du IBM® SPSS® Collaboration and Deployment Services Repository
- suppression des artefacts soumis expirés
- suppression des historiques de tâches

Vous pouvez appeler ces scripts depuis une étape de tâche générale dans IBM® SPSS® Collaboration and Deployment Services Deployment Manager pour effectuer des tâches de maintenance du référentiel.

Remarques

Ces informations ont été développées pour des produits et des services proposés dans le monde entier.

Il est possible qu'IBM ne propose pas les produits, services ou fonctionnalités présentés dans ce document dans d'autres pays. Consultez votre représentant IBM local pour des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, programme ou service IBM n'a aucunement l'intention d'affirmer ou de sous-entendre que seul ce produit, programme ou service IBM peut être utilisé. Tout produit, programme ou service de fonctionnalité équivalente et qui ne viole aucun droit de propriété intellectuelle IBM peut être utilisé. Cependant, il est de la responsabilité de l'utilisateur d'évaluer et de vérifier le fonctionnement de tout produit, programme ou service autre qu'IBM.

Il est possible qu'IBM dispose de brevets ou d'applications de brevet en attente qui couvrent le sujet décrit dans ce document. L'obtention de ce document ne vous accorde aucune licence associée à ces brevets. Vous pouvez envoyer vos questions sur les licences en écrivant à :

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Pour des questions sur les licences avec des ensembles de caractères double octets (DBCS), contactez le service de propriété intellectuelle IBM de votre pays ou envoyez vos questions par écrit à :

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

Le paragraphe suivant ne concerne pas le Royaume-Uni ou tout autre pays dans lesquels ces dispositions ne respectent pas le droit local : INTERNATIONAL BUSINESS MACHINES (IBM) FOURNIT LA PRESENTE PUBLICATION « TELLE QU'ELLE », SANS GARANTIE D'AUCUNE SORTE, EXPLICITE OU IMPLICITE, CE QUI COMPREND, SANS S'Y LIMITER, TOUTE GARANTIE IMPLICITE DE NON-CONTREFAÇON, DE COMMERCIALISATION OU DE CONVENANCE A UN USAGE PARTICULIER. Certains Etats ne permettent pas de renoncer à l'expression de garanties explicites ou implicites dans certaines transactions ; par conséquent, cette déclaration peut ne pas s'appliquer à vous.

Ces informations peuvent comporter des inexactitudes techniques ou des fautes d'orthographe. Des modifications sont fréquemment apportées aux informations présentées ici ; ces modifications seront intégrées dans de futures éditions de cette publication. IBM peut apporter à tout moment des améliorations et/ou des modifications au(x) produit(s) et/ou au(x) programme(s) décrit(s) dans cette publication et ce, sans préavis.

Toute référence incluse dans ces informations à des sites Web autres qu'IBM est fournie dans un but pratique uniquement et ne constitue pas une recommandation de ces sites Web. Le matériel de ces sites Web ne fait pas partie du matériel de ce produit IBM et l'utilisation de ces sites Web se fait à vos propres risques.

L'entreprise IBM a le droit d'utiliser ou de distribuer les informations que vous avez fournies de la façon dont elle le juge approprié sans être redevable d'aucune obligation envers vous.

Les détenteurs de licence de ce programme qui souhaitent obtenir des informations dans le but d'autoriser : (i) l'échange d'informations entre des programmes créés indépendamment et d'autres programmes (y compris celui-ci) et (ii) l'utilisation mutuelle des informations qui ont été échangées, doivent contacter :

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Ces informations peuvent être disponibles, conformément aux conditions générales appropriées, et soumises dans certains cas au paiement de frais.

Le programme sous licence décrit dans ce document et tout le matériel sous licence associé sont fournis par IBM conformément à l'accord client d'IBM, l'accord sur les licences de programme international d'IBM et d'autres accords équivalents.

Les données de performance contenues ici ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats obtenus dans d'autres environnements d'exploitation peuvent varier de manière significative. Il est possible que certaines mesures aient été effectuées sur des systèmes en cours de développement et rien ne garantit que ces mesures seront les mêmes sur des systèmes publics. De plus, certaines mesures peuvent avoir été obtenues par extrapolation. Les résultats réels peuvent varier. Les utilisateurs de ce document doivent vérifier les données applicables à leur environnement spécifique.

Les informations concernant des produits autres qu'IBM ont été obtenues auprès des fournisseurs de ces produits, des annonces qu'ils ont publiées ou d'autres sources publicitaires disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances, leur compatibilité ou toute autre fonctionnalité associée à des produits non-IBM. Les questions sur les performances de produits autres qu'IBM doivent être adressées aux fournisseurs de ces produits.

Toutes les déclarations concernant le futur ou les intentions d'IBM peuvent être modifiées ou retirées sans préavis et représentent uniquement des objectifs.

Ces informations contiennent des exemples de données et de rapports utilisés dans des opérations professionnelles quotidiennes. Pour les illustrer de manière aussi complète que possible, les exemples contiennent des noms d'individus, d'entreprises, de marques et de produits. Tous ces noms sont fictifs et toute similarité avec des noms et adresses utilisés par une entreprise réelle serait pure coïncidence.

Si vous lisez ces informations sur une copie papier, il est possible que les photographies et les illustrations en couleur n'apparaissent pas.

Trademarks

IBM, le logo IBM, ibm.com et SPSS sont des marques d'IBM Corporation, déposées dans de nombreuses juridictions internationales. Une liste actualisée des marques IBM est disponible sur le Web à l'adresse <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques déposées ou des marques de Adobe Systems Incorporated aux États-Unis et/ou dans d'autres pays.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium et Pentium sont des marques commerciales ou des marques déposées d'Intel Corporation ou de ses filiales aux États-Unis et dans d'autres pays.

Linux est une marque déposée de Linux Torvalds aux États-Unis, dans d'autres pays, ou les deux.

Microsoft, Windows et Windows NT et le logo Windows sont des marques commerciales de Microsoft Corporation aux États-Unis et/ou dans d'autres pays.

UNIX est une marque déposée de The Open Group aux États-Unis et dans d'autres pays.

Java et toutes les marques et logos Java sont des marques commerciales de Sun Microsystems, Inc. aux États-Unis et/ou dans d'autres pays.

D'autres noms de produits et de services peuvent être des marques d'IBM ou d'autres entreprises.



- ACL, 33, 41, 48
- Classe IdentificationSpecifier, 55
- Classe jobExecutionDetails, 61
- Classe jobStepChildExecutionDetails, 63
- Classe jobStepExecutionDetails, 62
- Classe PageResult, 32, 49, 55, 60–61
- Classe Resource, 40, 42–43, 45, 54
- Classe Row, 61
- Classe SearchChildRow, 56
- classe SearchRow, 55
- dossiers
 - copie, 35
 - création, 36
 - déplacement, 46
 - enfants de, 43
 - métadonnées, 44
 - suppression, 38
- éléments expirés
 - suppression, 64
- Étiquettes
 - application, 50
 - suppression, 48
- fichiers
 - copie, 35
 - déplacement, 46
 - exportation, 40
 - importation, 45
 - métadonnées, 44
 - suppression, 36
 - téléchargement, 39, 52
 - versions, 42
- historiques des tâches
 - suppression, 64
- IBM i, 3
- installation, 1
 - IBM i, 3
 - UNIX, 2
 - Windows, 1
- legal notices, 65
- listes de contrôle d'accès
 - application, 33
 - récupération, 41
 - suppression, 48
- méthode advanceSearch, 31
- méthode applySecurity, 33
- méthode cancelJob, 57
- méthode cascadeSecurity, 34
- méthode copyResource, 35
- méthode createFolder, 36
- méthode deleteFile, 36
- méthode deleteFileVersion, 37
- méthode deleteFolder, 38
- méthode deleteJobExecutions, 57
- méthode downloadFile, 39
- méthode executeJob, 58
- méthode exportResource, 40
- méthode getAccessControlList, 41
- méthode getAllVersions, 42
- méthode getChildren, 43
- méthode getCustomPropertyValue, 44
- méthode getJobExecutionDetails, 59
- méthode getJobExecutionList, 60
- méthode getMetadata, 44
- méthode importResource, 45
- méthode moveResource, 46
- méthode removeLabel, 48
- méthode removeSecurity, 48
- méthode search, 49
- méthode setLabel, 50
- méthode setMetadata, 51
- méthode uploadFile, 52
- module PESImpl, 30
 - exemples, 64
- opération advanceSearch, 7
- opération applySecurity, 9
- opération cascadeSecurity, 10
- opération copyResource, 10
- opération createFolder, 11
- opération deleteFile, 11
- opération deleteFileVersion, 12
- opération deleteFolder, 13
- opération deleteJobExecutions, 26
- opération downloadFile, 14
- opération executeJob, 26
- opération export, 15
- opération getAccessControlList, 15
- opération getAllVersions, 16
- opération getChildren, 16
- opération getCustomPropertyValue, 17
- opération getJobExecutionDetails, 27
- opération getJobExecutionList, 28
- opération getMetadata, 17
- opération import, 18
- opération moveResource, 19
- opération removeLabel, 20
- opération removeSecurity, 21
- opération search, 21
- opération setLabel, 22
- opération setMetadata, 23
- opération uploadFile, 24

recherche

options avancées, 31

SSL, 30

SSL (Secure Sockets Layer), 30

suppression

éléments expirés, 64

historiques des tâches, 64

tâches

annulation, 57

détails d'exécution, 59

exécution, 58

exécutions, 57, 60

trademarks, 66

UNIX, 2

versions de fichier

suppression, 37

Windows, 1