

IBM SPSS Collaboration and Deployment Services -
Essentials for Python
Version 6.0

Guide du développeur

IBM

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 53.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette version concerne la version 6, édition 0, modification 0 d'IBM SPSS Collaboration and Deployment Services et toutes les publications et modifications suivantes jusqu'à mention contraire dans les nouvelles versions.

© Copyright IBM Corporation 2000, 2013.

Table des matières

Avis aux lecteurs canadiens	v	Mots clés	20
		Opérations de gestion de processus	20
IBM SPSS Collaboration and Deployment Services - Essentials for Python	1	Le module PESImpl	23
Présentation	1	API de Référentiel de contenu	23
Installation	1	Méthodes	24
		Classes wrapper	44
Scripts de ligne de commande	3	API de gestion des processus	46
Mots clés globaux	3	Méthodes	46
Connexions du référentiel	3	Classes wrapper	49
Génération de scripts pour le référentiel de contenu	4	Exemples de scripts	52
Mots clés	4		
Opérations du référentiel de contenu	5	Remarques	53
Fonctions de gestion des processus	19	Marques	55

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

IBM SPSS Collaboration and Deployment Services - Essentials for Python

Présentation

IBM® SPSS Collaboration and Deployment Services fournit un canevas 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 travaux par lots combinant un ensemble de programmes, à utiliser avec les travaux et objets du référentiel. Ceci permet de simplifier considérablement les tâches en masse telles que :

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

Le canevas 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.

Installation

Le canevas de génération de scripts peut être installé sur les plateformes 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.

Pour obtenir des instructions d'installation, reportez-vous au guide d'installation IBM SPSS Collaboration and Deployment Services.

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 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 <Operation> <Keywords>
```

où :

- <Operation> 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

Le tableau 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. La distinction entre majuscules et minuscules doit être effectuée pour les mots clés.

Tableau 1. Mots clés globaux.

Mot clé	Version courte facultative	Utilisation
--user	-u	Nom d'utilisateur servant à la connexion au serveur de référentiel
--password	-p	Mot de passe servant à la connexion au serveur de référentiel
--host	-q	Nom de l'hôte/du serveur sur lequel le référentiel est installé
--port	-o	Numéro de port du serveur de référentiel
--ssl		Indique que le serveur de référentiel utilise le protocole SSL (Secure Sockets Layer) pour chiffrer les communications. Si vous utilisez ce mot clé, le serveur de référentiel doit être configuré pour SSL. Pour plus d'informations, reportez-vous à la documentation de l'administrateur.
--server_url	-S	Indique l'URL du serveur de référentiel. Utilisez ce mot clé si l'URL du serveur inclut une racine de contexte personnalisé. Aucune valeur n'est nécessaire pour les mots clés host, port et ssl si vous indiquez l'URL du serveur.
--useDefault	-z	Utilise les informations de connexion du serveur définies dans le fichier <code>Authorization.properties</code>
-h		Les informations d'aide du module de génération de scripts

Connexions du référentiel

Vous devez indiquer l'ID utilisateur, le mot de passe, l'hôte et les informations de serveur de référentiel d'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, comme dans l'exemple suivant :

```
--user user --password password --host host --port port
--user
utilisateur --password mot de passe
--server_url url
```

- Via le fichier `Authorization.properties`, où la commande contient un paramètre `--useDefault` (ou la version courte `-z`). Cette approche permet d'extraire les informations de connexion 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=mdp
host=yourhost
port=80
```

Vous pouvez également utiliser le mot clé `server_url` dans le fichier de propriétés.

```
# Authorization Information
user=admin
password=mdp
server_url=http://votrehte:80/racine_contexte
```

Les paramètres transmis via la ligne de commande sont toujours prioritaires. Par exemple, si `--user` et `--password` sont fournis sur 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 et l'hôte et le port sont extraits du fichier `Authorization.properties`. De plus, si les éléments `user`, `password`, `host` et `port` sont tous fournis sur la ligne de commande mais que le paramètre `--useDefault` ou `-z` est également utilisé, le paramètre `--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 minimal 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 libellés, 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

Le tableau 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.

Tableau 2. Mots clés des API du référentiel.

Mot clé	Version courte facultative	Utilisation
<code>--source</code>	<code>-s</code>	Le fichier source ou le chemin d'accès au dossier
<code>--target</code>	<code>-t</code>	Le chemin d'accès au dossier cible
<code>--version</code>	<code>-v</code>	La version d'un fichier

Tableau 2. Mots clés des API du référentiel (suite).

Mot clé	Version courte facultative	Utilisation
--principal	-r	L'utilisateur qui doit recevoir l'autorisation
--permission	-n	Le type d'autorisation (lecture, écriture, modification, suppression, par exemple)
--label	-l	Le libellé 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 principaux
--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>Travaux soumis</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

Pour toutes les opérations qui acceptent les informations sur le libellé et la version, l'utilisateur doit indiquer un libellé ou une version, mais pas les deux. Si aucune version ou aucun libellé n'est indiqué pour un fichier donné, la dernière version est utilisée.

Opérations du référentiel de contenu

Opération `advanceSearch`

Permet de rechercher des fichiers et des dossiers dans le référentiel, en fonction de différents paramètres.

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

Syntaxe

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

où :

- *<author>* (auteur) est le nom de l'auteur.
- *<title>* (titre) est le titre du fichier/dossier.
- *<description>* est la description du fichier/dossier.
- *<createdBy>* est le nom de l'utilisateur qui a créé le fichier/dossier.
- *<modifiedBy>* est le nom de l'utilisateur qui a modifié le fichier/dossier.
- *<keyword>* (mot-clé) est le mot clé associé au fichier/dossier.
- *<label>* (libellé) est le libellé du repère de version.
- *<topic>* (rubrique) est la rubrique associée au fichier/dossier.
- *<uri>* est l'URI du fichier. Le format d'URI est scpscr:///?id=xx. Veuillez noter que les dossiers ne sont pas inclus dans la recherche URI.
- *<parentURI>* est l'URI du dossier dans lequel effectuer la recherche. Le format d'URI est scpscr:///?id=xx.
- *<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, +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 Travaux soumis.

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>`

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 "<principal>"
--permission "<permission>" --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>` est l'utilisateur (*admin*, par exemple) à appliquer au fichier ou au dossier spécifié au sein 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.
- `<provider>` est le fournisseur de sécurité à utiliser pour la récupération des informations sur les utilisateurs (principaux). 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 :

- <permission> autorisation définie correctement pour <source>.
- <source> Aucun fichier ou dossier de ce type n'existe. Faites une nouvelle tentative.
- <permission> Type d'autorisation non valide. Veuillez recommencer.
- <source> Erreur lors de la définition de l'ACL de sécurité.

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. Faites une nouvelle tentative.
- <source> Erreur lors de la mise en cascade de l'ACL de sécurité.

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 «Opération moveResource», à la page 15 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. Faites une nouvelle tentative.
- <target> (cible) Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- <source> Erreur lors de la copie du fichier ou du dossier.

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. Faites une nouvelle tentative.
- <dossier> Le dossier existe déjà. Faites une nouvelle tentative.
- <source> Erreur lors de la création du dossier.

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 Travaux soumis. Ceci est un paramètre facultatif.

Exemple

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

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

L'exemple suivant permet de supprimer un fichier du dossier Travaux soumis, avec toutes ses versions^o:

```
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. Faites une nouvelle tentative.
- <source> Erreur lors de la suppression du fichier.

Opération deleteFileVersion

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

Syntaxe

```
python CADSTool.py deleteFileVersion --source "<source>" --version "<version>"  
--label "<label>" --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> (libellé) est le libellé du fichier à supprimer. Ceci est un paramètre facultatif.
- --submittedHierarchy permet de supprimer une version spécifique d'un fichier du dossier Travaux soumis. 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 un libellé spécifique :

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

L'exemple suivant permet de supprimer un fichier avec un libellé spécifique du dossier Travaux soumis :

```
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. Faites une nouvelle tentative.
- <source> Erreur lors de la suppression du fichier.

Opération deleteFolder

Permet de supprimer un dossier du référentiel, y compris l'intégralité de 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 Travaux soumis. 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 Travaux soumis :

```
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. Faites une nouvelle tentative.
- <source> Erreur lors de la suppression du dossier.

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 "<label>" --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> (libellé) est le libellé 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 repère de version^o:

```
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 libellé 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. Faites une nouvelle tentative.
- <target> (cible) Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- <source> Erreur lors du téléchargement du fichier.

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. Faites une nouvelle tentative.
- <source> Erreur lors l'exportation du dossier.

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. Faites une nouvelle tentative.
- Erreur lors de la récupération des détails de sécurité pour <source>.

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 Travaux soumis. 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 Travaux soumis :

```
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. Faites une nouvelle tentative.
- <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 au libellé et au repère de version.

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. Faites une nouvelle tentative.
- <source> Erreur lors de l'obtention des ressources.

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>.

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  
"<label>" --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>* (libellé) est le libellé du fichier ou du dossier pour lequel récupérer les métadonnées. Ceci est un paramètre facultatif.
- `--submittedHierarchy` permet de récupérer les métadonnées du dossier Travaux soumis. 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 libellé 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 libellé d'un fichier dans le dossier Travaux soumis :

```
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. Faites une nouvelle tentative.
- *<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.

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. Faites une nouvelle tentative.
- `<target>` Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- `<source>` Erreur lors de l'importation du dossier.

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 dans `Demo Folder`.
- **Cas 2** : Si le dossier `Demo Folder` n'existe pas, `Temp.txt` est déplacé dans `«/»` 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 dans `Demo Folder` et renommé en `Abc.dat`.
- **Cas 2** : Si le dossier `Demo Folder` n'existe pas, une erreur s'affiche.

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. Faites une nouvelle tentative.
- `<target>` (cible) Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- `<source>` Erreur lors du déplacement du fichier ou du dossier.

Opération removeLabel

Permet de supprimer un libellé d'un fichier dans le référentiel.

Syntaxe

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

où :

- *<source>* est le chemin d'accès complet du fichier dans le référentiel. Ceci est un paramètre obligatoire.
- *<label>* (libellé) est le nom du libellé 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 :

- Libellé supprimé avec succès pour *<source>*.
- *<source>* Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- *<source>* Erreur lors de la suppression du libellé.
- *<label>* Aucun libellé de ce type n'existe. Faites une nouvelle tentative.

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 "<principal>"  
--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>* est l'utilisateur/le principal (*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 (principaux). 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. Faites une nouvelle tentative.
- *<source>* Erreur lors de la suppression de l'ACL de sécurité.

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.
- *<criteria>* Aucun fichier ou dossier ne correspond aux critères de recherche.
- Erreur lors de la recherche des fichiers et des dossiers.

Opération setLabel

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

Syntaxe

```
python CADSTool.py setLabel --source "<source>" --version "<version>" --label "<label>" -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 le libellé. Ceci est un paramètre obligatoire.
- *<label>* (libellé) est le nom du libellé à appliquer à la version spécifiée du fichier. Ceci est un paramètre obligatoire.

Exemple

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

Messages

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

- Libellé correctement défini pour *<source>*.
- *<source>* Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- *<source>* Erreur lors de la définition du libellé.

Opération setMetadata

Permet d'appliquer les propriétés de métadonnées aux fichiers et aux dossiers dans le référentiel.

Le tableau 3 répertorie les propriétés de métadonnées et indique si elles peuvent être appliquées aux fichiers et/ou aux dossiers.

Tableau 3. Propriétés de métadonnées et types de ressources.

Propriété de métadonnées	Type de ressource
Auteur	Fichier
Description	Fichier ou dossier

Tableau 3. Propriétés de métadonnées et types de ressources (suite).

Propriété de métadonnées	Type de ressource
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
"<label>" --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, +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>* (libellé) est la version avec libellé 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. Faites une nouvelle tentative.
- <source> Erreur lors de la définition des métadonnées.

Opération 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 du fichier si 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. Faites une nouvelle tentative.
- <target> (cible) Aucun dossier de ce type n'existe. Faites une nouvelle tentative.
- <source> Erreur lors du téléchargement du fichier.

Fonctions de gestion des processus

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

- Exécution des travaux
- Extraction de l'historique des travaux
- Extraction des détails du travail

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

Le tableau 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 les mots clés supplémentaires qui s'appliquent à la fois aux API de gestion des processus et aux API du référentiel, voir tableau 1, à la page 3 et tableau 2, à la page 4.

Tableau 4. Mots clés des API de gestion des processus

Mot clé	Version courte facultative	Utilisation
--source	-s	Travail source, dont le chemin d'accès
--target	-t	Chemin d'accès au dossier cible
--notification	-j	Indique que le travail sera exécuté avec des notifications
--async	-m	Indique que le travail sera exécuté de façon asynchrone
--execId	-y	ID d'exécution du travail
--jobStepName	-q	Nom de l'étape du travail
--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 de gestion de processus

Opération deleteJobExecutions

Supprime les objets d'exécution de travail 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.

Opération executeJob

Exécute un travail 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 du travail. Pour une exécution asynchrone, l'API effectue un renvoi après le début du travail.

Syntaxe

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

où :

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

Exemples

L'exemple suivant permet d'exécuter le travail de façon synchrone sans notifications :

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

L'exemple suivant permet d'exécuter le travail de façon synchrone avec des notifications :

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

L'exemple suivant permet d'exécuter le travail de façon asynchrone sans notifications :

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

L'exemple suivant permet d'exécuter le travail 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>` Travail correctement exécuté. L'ID d'exécution du travail est `<execId>`.
- `<source>` Aucun travail de ce type n'existe. Faites une nouvelle tentative.
- `<source>` Erreur lors de l'exécution du travail.

Opération `getJobExecutionDetails`

Répertorie les détails d'exécution d'un travail spécifique, y compris les étapes et itérations du travail.

Syntaxe

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

où :

- `<execId>` est l'identificateur d'exécution du travail. Ceci est un paramètre obligatoire.
- `--log` indique que le journal des travaux doit être affiché en ligne. Si le paramètre `--log` n'est pas inclus, tout journal généré par une exécution d'étape de travail 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 travail spécifique :

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

L'exemple suivant décrit les détails d'une exécution de travail 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 travail spécifique, avec les journaux de travaux 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 le travail, les étapes, et les itérations du travail. 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. Faites une nouvelle tentative.
- `<execId>` Erreur lors de l'affichage des détails d'une exécution de travail.
- `--target` ne peut pas être utilisé sans le paramètre `--log`

Opération `getJobExecutionList`

Répertorie les exécutions en cours et terminées pour un travail spécifique, pour toutes les versions du travail.

Syntaxe

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

La valeur de `<source>` correspond au chemin complet du travail 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 du travail spécifiée, tous les détails d'exécution tels que l'ID d'exécution, le nom du travail, l'état d'exécution du travail, et l'heure de début et de fin de l'exécution du travail sont répertoriés.
- `<source>` Aucun travail de ce type n'existe. Faites une nouvelle tentative.
- `<source>` Erreur lors de l'affichage de la liste d'exécutions pour un travail.

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` correspond au nom de l'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` indique que le serveur de référentiel utilise le protocole secure sockets layer (SSL) pour chiffrer les communications. Si le paramètre `ssl` a la valeur `False` ou s'il est omis lors de la création de l'objet `PESImpl`, les communications du serveur n'utilisent pas SSL. Lorsque vous utilisez SSL, le serveur de référentiel doit être configuré pour cela. Pour plus d'informations, reportez-vous à la documentation de l'administrateur.

Vous pouvez également spécifier l'URL du serveur et non les paramètres `host`, `port` et `ssl`.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("utilisateur", "mot de passe",
                 server_url="url")
```

Le paramètre `server_url` spécifie l'URL complète pour le serveur de référentiel. Utilisez ce paramètre si votre serveur utilise un chemin contextuel personnalisé. Par exemple, le constructeur suivant correspond à un serveur nommé `myserver` qui utilise SSL sur le port 43 et dont le chemin contextuel est `/ibm/spss`:

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

Remarque : Toute adresse IPv6 doit être placée entre crochets (`[3ffe:2a00:100:7031::1]` par exemple).

Il est ensuite possible d'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 libellés, 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.

Remarque :

- 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 :

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

Méthode `advanceSearch`

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 :

- Auteur
- Description
- Titre
- Auteur de la création
- Modifié par
- Date de début de l'expiration
- Date de fin de l'expiration
- Type MIME
- Libellé
- Mot clé
- 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

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

Tableau 5. Paramètres d'entrée pour *advanceSearch*.

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

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

Tableau 6. Valeur de retour pour `advanceSearch`.

Type	Description
PageResult	La structure de chaque ligne correspond à une correspondance de la recherche. Pour plus d'informations, voir la rubrique «Classe PageResult», à la page 45.

Tableau 7. Exceptions pour `advanceSearch`.

Type	Description
InsufficientParameterException	Les paramètres obligatoires ne sont pas indiqués.

Exemple : Recherche par libellé et mot clé

L'exemple suivant renvoie toutes les versions de fichier avec libellé *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()
```

Méthode `applySecurity`

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

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

Tableau 8. Paramètres d'entrée d'`applySecurity`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>principal</i>	Requis	Chaîne	Utilisateur (tel que <i>admin</i>) à appliquer au fichier ou au dossier indiqué dans le cadre de l'ACL	admin

Tableau 8. Paramètres d'entrée d'applySecurity (suite).

Champ	Utilisation	Type	Description	Exemple de valeur
<i>permission</i>	Requis	Chaîne	Type d'autorisation à appliquer au fichier ou au dossier indiqué	READ, WRITE, DELETE, MODIFY_ACL OU OWNER
<i>provider</i>	Facultatif	Chaîne	Fournisseur de sécurité à utiliser pour appliquer la sécurité aux utilisateurs (<i>Natif</i> , par exemple)	Natif
<i>cascade</i>	Facultatif	Booléen	Propage les paramètres de sécurité à tous les fichiers et sous-dossiers du dossier indiqué	True ou False

Tableau 9. Valeur de retour pour applySecurity.

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

Tableau 10. Exceptions pour applySecurity.

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

Exemple

L'exemple suivant affecte à un utilisateur l'autorisation *READ* pour le fichier désigné.

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

Méthode cascadeSecurity

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

```
cascadeSecurity(source)
```

Tableau 11. Paramètres d'entrée de cascadeSecurity.

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

Tableau 12. Valeur de retour pour cascadeSecurity.

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

Tableau 13. Exceptions pour cascadeSecurity.

Type	Description
ResourceNotFoundException	Le dossier source n'existe pas.

Tableau 13. Exceptions pour `cascadeSecurity` (suite).

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

Exemple

L'exemple suivant répercute la sécurité du dossier *Projects* au contenu du dossier.

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

Méthode `copyResource`

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 l'attribution de nouveau nom, voir «Méthode `moveResource`», à la page 37.

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

Tableau 14. Paramètres d'entrée de `copyResource`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	Requis	Chaîne	Chemin d'accès complet ou 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"

Tableau 15. Valeur de retour pour `copyResource`.

Type	Description
Chaîne	URI du fichier ou du dossier copié

Tableau 16. 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
```

Méthode `createFolder`

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

```
createFolder(source)
```

Tableau 17. Paramètres d'entrée de `createFolder`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	Chaîne	Dossiers à créer dans le référentiel	/Temp Folder/Sample Folder

Tableau 18. Valeur de retour pour `createFolder`.

Type	Description
Chaîne	URI du dossier créé

Tableau 19. Exceptions pour `createFolder`.

Type	Description
<code>InsufficientParameterException</code>	Les paramètres requis ne sont pas indiqués.
<code>ResourceAlreadyExistsException</code>	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."
```

Méthode `deleteFile`

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

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

Tableau 20. Paramètres d'entrée de `deleteFile`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	Facultatif	Booléen	Indique si le fichier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 21. Valeur de retour pour `deleteFile`.

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

Tableau 22. 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.PESExceptions 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."
```

Méthode deleteFileVersion

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

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

Tableau 23. Paramètres d'entrée de deleteFileVersion.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>version</i>	Facultatif. Cependant, <i>version</i> ou <i>libellé</i> doit être spécifié.	Chaîne	La version spécifique du fichier à supprimer	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facultatif. Cependant, <i>version</i> ou <i>libellé</i> doit être spécifié.	Chaîne	La version spécifique avec libellé du fichier à supprimer	"Version 1"
<i>submittedHierarchy</i>	Facultatif	Booléen	Indique si le fichier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 24. Valeur de retour pour deleteFileVersion.

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

Tableau 25. Exceptions pour deleteFileVersion.

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

```

Méthode deleteFolder

Permet de supprimer un dossier et tout son contenu du référentiel

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

Tableau 26. Paramètres d'entrée de deleteFolder.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du dossier dans le référentiel	"/Temp Folder" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>submittedHierarchy</i>	Facultatif	Booléen	Indique si le dossier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 27. Valeur de retour pour deleteFolder.

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

Tableau 28. 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."

```

Méthode downloadFile

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

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

Tableau 29. Paramètres d'entrée de `downloadFile`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet au référentiel ou URI d'objet du fichier à télécharger	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	Requis	Chaîne	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 le libellé peut être indiqué.	Chaîne	Version spécifique du fichier à télécharger	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facultatif. La version ou le libellé peut être indiqué.	Chaîne	Version spécifique avec libellé du fichier à télécharger	"Version 2"

Tableau 30. Valeur de retour pour `downloadFile`.

Type	Description
Resource	Conteneur d'informations concernant un objet du référentiel. Pour plus d'informations, voir la rubrique «Classe Resource», à la page 44.

Tableau 31. Exceptions pour `downloadFile`.

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 télécharge une version avec libellé *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")
```

Méthode `exportResource`

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

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

Tableau 32. Paramètres d'entrée de `exportResource`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet au référentiel ou URI d'objet du dossier à exporter	Dossier /Temp ou 0a58c3670016a7860000010dcee0eaa28219

Tableau 32. Paramètres d'entrée d'exportResource (suite).

Champ	Utilisation	Type	Description	Exemple de valeur
<i>target</i>	Requis	Chaîne	Chemin d'accès complet (dans le système de fichiers local) et nom de fichier vers lequel exporter le dossier	C:\Temp\backup.pes

Tableau 33. Valeur de retour pour exportResource.

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

Tableau 34. 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")
```

Méthode getAccessControlList

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

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

Tableau 35. Paramètres d'entrée de getAccessControlList.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a786000010dcee0eaa28219"
<i>submittedHierarchy</i>	Facultatif	Booléen	Indique si le fichier ou le dossier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 36. Valeur de retour pour getAccessControlList.

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

Tableau 37. 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
```

Méthode `getAllVersions`

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

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

Tableau 38. Paramètres d'entrée de `getAllVersions`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier dans le référentiel	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	Facultatif	Booléen	Indique si le fichier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 39. Valeur de retour pour `getAllVersions`.

Type	Description
List	Liste des objets de ressource. Consultez «Classe Resource», à la page 44.

Tableau 40. 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 repère de version et les libellés 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()
```

Méthode getChildren

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

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

Tableau 41. Paramètres d'entrée de getChildren.

Zone	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou 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 Travaux soumis	True ou False

Tableau 42. Valeur de retour pour getChildren.

Type	Description
List	Liste des objets de ressource. Consultez «Classe Resource», à la page 44.

Tableau 43. Exceptions pour getChildren.

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

Exemple

L'exemple suivant permet d'extraire le contenu du dossier */Demo/Drafts*, en imprimant le titre, l'auteur et l'identificateur de ressources pour chacun d'entre 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()
```

Méthode getCustomPropertyValue

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

```
getCustomPropertyValue(propertyName)
```

Tableau 44. Paramètres d'entrée de getCustomPropertyValue.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>propertyName</i>	Requis	Chaîne	Nom de la propriété personnalisée	"FreeForm"

Tableau 45. Valeur de retour pour getCustomPropertyValue.

Type	Description
List	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, String, Date ou Number).

Tableau 46. 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
```

Méthode `getMetadata`

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.

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

Tableau 47. Paramètres d'entrée de `getMetadata`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<code>version</code>	Facultatif. La version ou le libellé peut être indiqué.	Chaîne	Version spécifique du fichier ou du dossier	"0:2006-08-25 21:15:49.453"
<code>label</code>	Facultatif. La version ou le libellé peut être indiqué.	Chaîne	Version spécifique avec libellé du fichier ou du dossier	"Version 1"
<code>submittedHierarchy</code>	Facultatif	Booléen	Indique si le fichier figure dans le dossier <i>Travaux soumis</i>	True ou False

Tableau 48. Valeur de retour pour `getMetadata`.

Type	Description
Resource	Conteneur d'informations concernant un objet du référentiel. Pour plus d'informations, voir la rubrique «Classe Resource», à la page 44

Tableau 49. Exceptions pour `getMetadata`.

Type	Description
<code>ResourceNotFoundException</code>	Le fichier ou le dossier source n'existe pas.
<code>InsufficientParameterException</code>	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()

```

Méthode importResource

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

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

Tableau 50. Paramètres d'entrée d'importResource.

Champ	Utilisation	Type	Description	Exemple de valeur
source	Requis	Chaîne	Chemin d'accès complet (dans le système de fichiers local) du fichier à importer	"C:\Temp\New.pes"
target	Requis	Chaîne	Chemin d'accès complet au référentiel ou URI d'objet du dossier dans lequel réaliser l'importation	"/Temp Folder" ou "0a58c3670016a7860000010dcee0eaa28219"

Tableau 51. Valeur de retour pour importResource.

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

Tableau 52. 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 importe le contenu du fichier drafts.pes dans le dossier /Demo/Drafts du référentiel IBM SPSS Collaboration and Deployment Services Repository

```

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

```

Méthode moveResource

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 :

Tableau 53. 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.

`moveResource(source, target)`

Tableau 54. Paramètres d'entrée de `moveResource`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>source</code>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	<code>/Temp Folder/Temp.txt</code> ou <code>0a58c3670016a7860000010dcee0eaa28219</code>
<code>target</code>	Requis	Chaîne	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.	Dossier <code>/New</code> ou <code>/New Folder/abc.dat</code>

Tableau 55. Valeur de retour pour `moveResource`.

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

Tableau 56. 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 déplace le fichier `MyReport.rptdesign` du dossier `/Demo/Drafts` vers le dossier `/Approved`.

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

Méthode `removeLabel`

Permet de supprimer un libellé d'un fichier dans le référentiel.

`removeLabel(source, label)`

Tableau 57. Paramètres d'entrée de `removeLabel`.

Champ	Utilisation	Type	Exemple de valeur	Description
<i>source</i>	Requis	Chaîne	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"	Chemin d'accès complet ou URI de l'objet du fichier dans le référentiel
<i>label</i>	Requis	Chaîne	"Version 1"	Nom de libellé à supprimer

Tableau 58. Valeur de retour pour `removeLabel`.

Type	Description
String	URI du fichier mis à jour

Tableau 59. Exceptions pour `removeLabel`.

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

Exemple

L'exemple suivant permet de supprimer le libellé *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")
```

Méthode `removeSecurity`

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

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

Tableau 60. Paramètres d'entrée de `removeSecurity`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou URI de l'objet du fichier ou du dossier dans le référentiel.	"/Temp Folder/Temp.txt" ou "0a58c3670016a7860000010dcee0eaa28219"
<i>principal</i>	Requis	Chaîne	Utilisateur (<i>admin</i> , par exemple) à supprimer du fichier ou du dossier indiqué	admin
<i>provider</i>	Facultatif	Chaîne	Fournisseur de sécurité (<i>Native</i> , par exemple) à utiliser pour obtenir les informations sur les utilisateurs	Natif
<i>cascade</i>	Facultatif	Booléen	Propage les paramètres de sécurité à tous les fichiers et sous-dossiers du dossier indiqué	True ou False

Tableau 61. Valeur de retour pour `removeSecurity`.

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

Tableau 62. 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) pour un principal 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")
```

Méthode `search`

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.

```
search(criteria)
```

Tableau 63. Paramètres d'entrée de `search`.

Champ	Utilisation	Type	Description	Exemple de valeur
<code>criteria</code>	Requis	Chaîne	Valeur permettant de rechercher les métadonnées de fichiers	"Age"

Tableau 64. Valeur de retour pour `search`.

Type	Description
<code>PageResult</code>	La structure de chaque ligne correspond à une correspondance de la recherche. Pour plus d'informations, voir la rubrique «Classe <code>PageResult</code> », à la page 45.

Tableau 65. Exceptions pour `search`.

Type	Description
<code>InsufficientParameterException</code>	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()
```

Méthode setLabel

Permet d'appliquer un libellé à une version d'un fichier dans le référentiel. Si le fichier comporte déjà un libellé, celui d'origine est remplacé par le nouveau.

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

Tableau 66. Paramètres d'entrée de setLabel.

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

Tableau 67. Valeur de retour pour setLabel.

Type	Description
String	URI du fichier mis à jour

Tableau 68. 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 le libellé *Beta* à la seconde version du fichier *MyReport.rptdesign*. La méthode `getVersionMarker` pour un objet `Resource` renvoie le repère 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")
```

Méthode setMetadata

Permet d'appliquer les propriétés de métadonnées aux fichiers et aux dossiers dans le référentiel.

Le tableau suivant indique les propriétés de métadonnées et précise si ces dernières peuvent être appliquées aux fichiers et aux dossiers.

Tableau 69. 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

Tableau 69. Propriétés des objets du référentiel (suite).

Propriété de métadonnées	Type de ressource
Métadonnées personnalisées	Fichier ou dossier

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

Tableau 70. Paramètres d'entrée de `setMetadata`.

Zone	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet ou 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 le libellé peut être indiqué.	Chaîne	La version spécifique du fichier à télécharger	"0:2006-08-25 21:15:49.453"
<i>label</i>	Facultatif. La version ou le libellé peut être indiqué.	Chaîne	Libellé 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 <code>topic</code> et <code>Dictionnaire</code> comme <code>customProperty</code> . Pour le reste des métadonnées, il utilise une chaîne.	{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }

Tableau 71. Valeur de retour pour `setMetadata`.

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

Tableau 72. 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
```

Méthode uploadFile

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.

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

Tableau 73. Paramètres d'entrée d'uploadFile.

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

Tableau 74. Valeur de retour pour uploadFile.

Type	Description
String	URI du fichier transféré

Tableau 75. 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 versionFlag 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.

Classe Resource

La classe Resource sert de wrapper simplifié au l'objet de référentiel ResourceSpecifer.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. Le tableau 76 répertorie toutes les méthodes disponibles dans la classe Resource.

Tableau 76. Méthodes pour la classe Resource.

Nom de la méthode	Description
getAccessControlList	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 <i>ressource X</i> , alors <code>getAccessControlList</code> de l'objet <i>resource</i> représentant <i>X</i> renvoie <code>{ 'Joe' : 'DELETE' }</code> et non les trois autorisations (lecture, écriture, suppression) de l'appel de service Web.
getOwner	Renvoie le nom du propriétaire de l'objet sous forme de chaîne
getAuthor	Renvoie le nom de l'auteur de l'objet sous forme de chaîne
getContentSize	Renvoie la taille de l'objet
getCreatedBy	Renvoie le nom de l'utilisateur qui a créé l'objet sous forme de chaîne
getCreationDate	Renvoie la date de création de l'objet sous forme d'objet datetime
getDescription	Renvoie la description de l'objet sous forme de liste
getDescriptionLanguage	Renvoie la langue de l'objet sous forme de liste
getExpirationDate	Renvoie la date d'expiration de l'objet sous forme d'objet datetime
isExpired	Indique si l'objet spécifié a expiré ou non
getMIMEType	Renvoie le type MIME de l'objet sous forme de chaîne
getModificationDate	Renvoie la date de dernière modification de l'objet sous forme d'objet datetime
getObjectCreationDate	Renvoie la date de création de l'objet sous forme d'objet datetime
getObjectLastModifiedBy	Renvoie l'utilisateur qui a modifié pour la dernière fois l'objet sous forme de chaîne
getObjectLastModifiedDate	Renvoie la date de dernière modification de l'objet sous forme d'objet datetime
getResourceID	Renvoie l'identificateur de ressources de l'objet sous forme de chaîne
getResourcePath	Renvoie le chemin d'accès de l'objet spécifié sous forme de chaîne
getTitle	Renvoie le titre de l'objet sous forme de chaîne
getTopicList	Renvoie la liste de rubriques pour l'objet
getVersionMarker	Renvoie la version de l'objet sous forme de chaîne.
getVersionLabel	Renvoie le libellé de l'objet sous forme de chaîne
getCustomMetadata	Renvoie des propriétés personnalisées associées à l'objet sous forme de dictionnaire

Tableau 76. Méthodes pour la classe Resource (suite).

Nom de la méthode	Description
getKeywordList	Renvoie une liste de mots-clés associés à l'objet

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.

Le tableau 77 répertorie toutes les méthodes disponibles dans la classe IdentificationSpecifieur.

Tableau 77. Méthodes pour la classe IdentificationSpecifieur.

Nom de la méthode	Description
getIdentifieur	Renvoie la valeur de l'identificateur d'un objet sous forme de chaîne
getVersionMarker	Renvoie la version d'un l'objet sous forme de chaîne
getVersionLabel	Renvoie le libellé appliqué à un objet sous forme de chaîne

Classe PageResult

Cette classe PageResult sert de conteneur pour les résultats de la recherche. Une occurrence 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. Le tableau 78 répertorie toutes les méthodes disponibles dans la classe PageResult.

Tableau 78. Méthodes pour la classe PageResult.

Nom de la méthode	Description
getRows	Renvoie une liste d'objets SearchRow. Pour plus d'informations, voir la rubrique «Classe SearchRow».

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.

Le tableau 79 répertorie toutes les méthodes disponibles dans la classe SearchRow.

Tableau 79. Méthodes pour la classe SearchRow.

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

Tableau 79. Méthodes pour la classe SearchRow (suite).

Nom de la méthode	Description
getChildRow	Renvoie la liste des objets SearchChildRow (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.

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.

Le tableau 80 répertorie toutes les méthodes disponibles dans la classe SearchChildRow.

Tableau 80. 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 le libellé 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 repère 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 travaux. Cette zone comprend les fonctionnalités suivantes :

- Exécution des travaux
- Extraction de l'historique des travaux
- Extraction des détails du travail

Cette section décrit les méthodes PESImpl utilisées pour la manipulation des travaux stockés 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.

Méthode cancelJob

Permet d'annuler un travail en cours.

```
cancelJob(executionId)
```

Tableau 81. Paramètres d'entrée de cancelJob.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>executionId</i>	Requis	Chaîne	ID d'exécution du travail	0a58c33d002ce90800 00010e0ccf7b01800e

Tableau 82. Valeur de retour pour cancelJob.

Type	Description
Boolean	Renvoie un message de succès lors de l'annulation du travail

Exemple

Cet exemple présente l'arrêt de l'exécution du travail *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
```

Méthode deleteJobExecutions

Supprime une ou plusieurs exécutions de travaux

```
deleteJobExecutions(executionId)
```

Tableau 83. Paramètres d'entrée de deleteJobExecutions.

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

Tableau 84. Valeur de retour 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 du travail *Reports*.

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

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

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

Méthode executeJob

Exécute un travail 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 du travail. Pour une exécution asynchrone, la méthode effectue un renvoi après le début du travail.

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

Tableau 85. Paramètres d'entrée de la méthode executeJob.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet (dans le système de fichiers local) du fichier à transférer	"C:\Temp\Temp.txt"
<i>notification</i>	Facultatif	Booléen	Indique si le travail est exécuté avec ou sans notifications. La valeur par défaut est False.	True ou False
<i>asynchronous</i>	Facultatif	Booléen	Indique si le travail est exécuté ou non de façon asynchrone. La valeur par défaut est False.	True ou False

Tableau 86. Valeur de retour pour executeJob.

Type	Description
String	Identifiant unique de l'exécution du travail. Cet identifiant est utilisé comme référence à l'exécution d'une d'un travail particulier.

Exemple

Cet exemple présente le début de l'exécution du travail *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
```

Méthode getJobExecutionDetails

Répertorie les détails d'exécution d'un travail spécifique, notamment les étapes et itérations du travail

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

Tableau 87. Paramètres d'entrée de getJobExecutionDetails.

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

Tableau 88. Valeur de retour pour getJobExecutionDetails.

Type	Description
jobExecutionDetails	Détails sur l'exécution d'un travail. Pour plus d'informations, voir la rubrique «Classe jobExecutionDetails», à la page 50.

Exemple

Cet exemple présente la récupération des informations sur les exécutions des étapes du travail 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()
```

Méthode getJobExecutionList

Permet de répertorier les exécutions d'un travail spécifique, dont des travaux en cours et terminés, pour toutes les versions du travail.

`getJobExecutionList(source)`

Tableau 89. Paramètres d'entrée de `getJobExecutionList`.

Champ	Utilisation	Type	Description	Exemple de valeur
<i>source</i>	Requis	Chaîne	Chemin d'accès complet du travail dans le référentiel.	"/testJob"

Tableau 90. Valeur de retour pour `getJobExecutionList`.

Type	Description
PageResult	Conteneur de la liste des exécutions d'un travail. Pour plus d'informations, voir la rubrique «Classe PageResult». .

Exemple

Cet exemple présente la récupération des exécutions du travail *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.

Classe PageResult

Cette classe PageResult sert de conteneur pour les résultats d'exécution du travail, permettant de récupérer les données spécifiques à l'exécution du travail.

Une exécution de travail individuel correspond à une ligne de l'objet `PageResult`. Par exemple, un travail exécuté quatre fois correspond à un objet `PageResult` contenant quatre lignes. Le tableau 91 répertorie toutes les méthodes disponibles dans la classe `PageResult`.

Tableau 91. 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'un travail. Pour plus d'informations, voir la rubrique «Classe <code>Row</code> ».

Classe `Row`

La classe `Row` sert de conteneur aux informations de niveau de travail, relatives à l'exécution d'un travail. Vous pouvez accéder aux métadonnées relatives à l'exécution d'un travail à l'aide des méthodes de cette classe.

Le tableau 92 répertorie toutes les méthodes disponibles dans la classe `Row`.

Tableau 92. Méthodes pour la classe `Row`.

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

Classe `jobExecutionDetails`

Cette classe est renvoyée depuis la méthode `getJobExecutionDetails`. Elle stocke les détails de l'exécution d'un travail et inclut une liste d'objets `jobStepExecution` fournissant des informations sur chaque étape du travail.

Le tableau 93 répertorie toutes les méthodes disponibles dans la classe `jobExecutionDetails`.

Tableau 93. 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, voir la rubrique «Classe <code>jobStepExecutionDetails</code> », à la page 51.
<code>getArtifactLocation</code>	Renvoie une liste des emplacements d'artefacts du travail
<code>getCompletionCode</code>	Renvoie le code d'achèvement de l'exécution du travail
<code>getEndDateTime</code>	Renvoie la date et l'heure de fin de l'exécution du travail
<code>getEventName</code>	Renvoie le nom d'événement de l'exécution du travail
<code>getEventUUID</code>	Renvoie l'ID d'événement de l'exécution du travail
<code>getExecutionState</code>	Renvoie l'état de l'exécution du travail

Tableau 93. Méthodes pour la classe `jobExecutionDetails` (suite).

Nom de la méthode	Description
<code>getExecutionSuccess</code>	Renvoie l'état de réussite ou d'échec de l'exécution du travail
<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 email sont activées ou non
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'exécution du travail
<code>getStartDateTime</code>	Renvoie la date et l'heure de début de l'exécution du travail
<code>getUserName</code>	Renvoie le nom de l'utilisateur qui a exécuté le travail
<code>getUUID</code>	Renvoie l'ID d'exécution du travail

Classe `jobStepExecutionDetails`

Cette classe stocke les détails d'exécution d'une étape de travail ainsi qu'une liste des objets `jobStepChildExecutionDetails`. Cette classe contient l'objet `ExecutionDetails`, auquel il délègue tous ses appels de méthodes.

Le tableau 94 répertorie toutes les méthodes disponibles dans la classe `jobStepExecutionDetails`.

Tableau 94. Méthodes pour la classe `jobStepExecutionDetails`.

Nom de la méthode	Description
<code>getJobStepChildExecutionList</code>	Renvoie une liste d'objets <code>jobStepChildExecutionDetails</code> . Pour plus d'informations, voir la rubrique «Classe <code>jobStepChildExecutionDetails</code> », à la page 52.
<code>getArtifactLocation</code>	Renvoie une liste des emplacements d'artefacts de l'étape du travail
<code>getCompletionCode</code>	Renvoie le code d'achèvement de l'étape du travail
<code>getEndTime</code>	Renvoie la date et l'heure de fin de l'étape du travail
<code>getEventName</code>	Renvoie le nom d'événement de l'étape du travail
<code>getEventUUID</code>	Renvoie l'ID d'événement de l'étape du travail
<code>getExecutionState</code>	Renvoie l'état d'exécution de l'étape du travail
<code>getExecutionSuccess</code>	Renvoie l'état de réussite ou d'échec de l'étape du travail
<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 email sont activées ou non
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'étape du travail
<code>getStartDateTime</code>	Renvoie la date et l'heure de début de l'étape du travail
<code>getUserName</code>	Renvoie le nom de l'utilisateur qui a exécuté l'étape du travail
<code>getUUID</code>	Renvoie l'ID d'exécution de l'étape du travail

Classe jobStepChildExecutionDetails

La classe `jobStepChildExecutionDetails` sert de conteneur aux exécutions enfants des étapes de travaux individuelles. Par exemple, une étape de travail 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.

Le tableau 95 répertorie toutes les méthodes disponibles dans la classe `jobStepChildExecutionDetails`.

Tableau 95. 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>getEndTime</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 email sont activées
<code>getQueuedDateTime</code>	Renvoie la date et l'heure de mise en file d'attente de l'exécution enfant
<code>getStartTime</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 :

`<emplacement installation>/samples`

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

- Suppression des éléments arrivés à expiration du référentiel IBM SPSS Collaboration and Deployment Services Repository
- Suppression des artefacts soumis arrivés à expiration
- Suppression des historiques de travaux

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

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. L'obtention de ce document ne vous accorde aucune licence associée à ces brevets. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

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

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7 Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

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

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAULT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent document contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programmes sont fournis "EN L'ETAT", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable de tout dommage lié à l'utilisation de ces exemples de programmes.

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

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp., aux Etats-Unis et/ou dans certains autres pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web «Copyright and trademark information» à 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 Etats-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 d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

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

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

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

