

IBM SPSS Collaboration and Deployment Services -
Essentials for Python
Versión 8 Release 0

Guía del desarrollador

IBM

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de "Avisos" en la página 63.

Información del producto

Esta edición se aplica a la versión 8, release 0, modificación 0 de IBM SPSS Collaboration and Deployment Services y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en ediciones nuevas.

© Copyright IBM Corporation 2000, 2016.

Contenido

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

Descripción general	1
Instalación de	1
Características en desuso	1

Capítulo 2. Scripts de línea de comandos 3

Palabras clave globales	3
Conexiones de repositorio	4
Scripts de repositorio de contenido	5
Palabras clave	5
Operaciones del repositorio de contenido	6
Funciones de gestión de procesos	22

Palabras clave	22
Operaciones de gestión de procesos	22

Capítulo 3. El módulo PESImpl 27

API del repositorio de contenido	28
Métodos	28
Clases de derivador	53
API de gestión de procesos	56
Métodos	56
Clases de derivador	59
Scripts de ejemplo	62

Avisos 63

Consideraciones sobre la política de privacidad	64
Marcas comerciales	65

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

Descripción general

IBM® SPSS Collaboration and Deployment Services proporciona un marco de scripts con un conjunto de API que tanto usuarios avanzados como administradores pueden utilizar para escribir rutinas independientes o trabajos por lotes que combinen un conjunto de rutinas para trabajar con objetos de repositorio y trabajos. Esto puede simplificar de forma considerable las tareas masivas, incluyendo las siguientes:

- Cambio de permisos de seguridad para un grupo grande de usuarios
- Etiquetado o eliminación de etiquetas de un gran número de carpetas o archivos
- Carga o descarga de muchas carpetas o archivos

El marco incluye la capacidad de realizar tareas de la línea de comandos, así como una rica API para interactuar con el Repositorio de IBM SPSS Collaboration and Deployment Services en su propio código Python.

Si desea información general sobre Python, un lenguaje de programación orientado a objetos dinámicos, consulte el sitio web de Python.

Instalación de

La infraestructura de scripts se puede instalar en las plataformas Windows y UNIX. La plataforma de scripts es independiente a la plataforma utilizada por el repositorio al que accede el recurso de scripts.

Por ejemplo, un repositorio que se ejecuta en la plataforma Windows se puede llamar mediante funciones de scripts que se ejecutan en la plataforma UNIX.

Para obtener instrucciones de instalación, consulte la Guía de instalación de IBM SPSS Collaboration and Deployment Services - Essentials for Python.

Características en desuso

Si está migrando de un release anterior de IBM SPSS Collaboration and Deployment Services, debe conocer las diversas características que han quedado en desuso desde la última versión.

Si una característica está en desuso, es posible que IBM Corp. elimine esta característica en un release subsiguiente del producto. La inversión futura se centrará en la función estratégica listada en la acción de migración recomendada. Normalmente, una característica no pasa a estar en desuso a menos que se proporcione una alternativa equivalente.

Las tablas siguientes indican qué está en desuso. Donde es posible, la tabla también indica la acción de migración recomendada.

Tabla 1. Características en desuso en versiones anteriores

Desuso	Acción de migración recomendada
Proveedor de seguridad: Active Directory con alteración temporal local, que soporta grupos extendidos y usuarios autorizados	Utilice el proveedor de seguridad estándar de Active Directory con los grupos necesarios añadidos

Tabla 1. Características en desuso en versiones anteriores (continuación)

Desuso	Acción de migración recomendada
IBM SPSS Collaboration and Deployment Services Enterprise View	Utilice la característica de la vista de datos analíticos
IBM SPSS Collaboration and Deployment Services Enterprise View Driver	Utilice la característica de la vista de datos analíticos
Instalación de web para IBM SPSS Collaboration and Deployment Services Deployment Manager	Utilice el instalador autónomo
BIRT Report Designer for IBM SPSS	Ninguna
Visor de BIRT Report Designer for IBM SPSS	Ninguna
IBM SPSS Collaboration and Deployment Services Portlet	Utilice el IBM SPSS Collaboration and Deployment Services Deployment Portal directamente o utilice las API de servicios web
Componente web de IBM SPSS Collaboration and Deployment Services	Utilice el IBM SPSS Collaboration and Deployment Services Deployment Portal directamente o utilice las API de servicios web
API de Servicio de puntuación V1	API de Servicio de puntuación V2
Servicio de servidor de planificación	Ninguna
Servicio de creación de informes	Ninguna
Operación login de Servicio de autenticación	Operación doLogin de Servicio de autenticación
Operación search de Servicio de búsqueda	Operación search2.5 de Servicio de búsqueda
jar de cliente de servicios web SPSS AXIS/Castor	Utilice las herramientas proporcionadas con Java Runtime Environment, Integrated Development Environment o Eclipse Web Tools Platform (WTP)

Para obtener información actualizada sobre las características en desuso, consulte IBM Knowledge Center.

Capítulo 2. Scripts de línea de comandos

El archivo de Python `CADSTool.py` puede usarse desde la línea de comandos para manipular los recursos almacenados en el Repositorio de IBM SPSS Collaboration and Deployment Services.

La sintaxis general usada para llamar las operaciones de scripts de IBM SPSS Collaboration and Deployment Services desde la línea de comandos es:

```
python CADSTool.py <Operación> <palabrasClave>
```

Donde:

- <Operación> designa la función a invocar
- <palabrasClave> define los pares de palabra clave/valor que utilizan como parámetros de entrada a la función

Palabras clave globales

Tabla 2 enumera las palabras clave compatibles con todas las funciones de scripts de IBM SPSS Collaboration and Deployment Services. La segunda columna enumera cualquier versión opcional y abreviada de las palabras clave. Las palabras clave distinguen entre mayúsculas y minúsculas.

Tabla 2. Palabras clave globales.

Palabra clave	Versión abreviada opcional	Uso
--user	-u	<p>El nombre de usuario para conectar con el servidor del repositorio. El valor debe incluir un prefijo que indica el proveedor de seguridad si el usuario no procede del proveedor nativo. Son válidos los valores de prefijo siguientes:</p> <ul style="list-style-type: none">• Native para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado.• AD_<nombre> para Active Directory, donde <nombre> corresponde al nombre de proveedor de seguridad dentro del sistema• ADL_<nombre> para Active Directory con alteración temporal local, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema• ldap_<nombre> para OpenLDAP, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema <p>Tras el prefijo, coloque una barra inclinada y el nombre de usuario. Para proveedores de Active Directory, incluya el dominio en el prefijo. Por ejemplo, para el usuario <i>icrod</i> del dominio <i>MYDOMAIN</i> de la instancia de Active Directory <i>AD_SPSSAD</i>, el valor es <i>AD_SPSSAD/MYDOMAIN/icrod</i>. Si el usuario <i>icrod</i> está en el proveedor de OpenLDAP <i>SPSSLDAP</i>, el valor será <i>ldap_SPSSLDAP/icrod</i>.</p>
--password	-p	La contraseña para conectar con el servidor del repositorio
--host	-q	El nombre de host/servidor donde está instalado el repositorio

Tabla 2. Palabras clave globales (continuación).

Palabra clave	Versión abreviada opcional	Uso
--port	-o	El número de puerto del servidor del repositorio
--ssl		Indica que el servidor del repositorio utiliza el protocolo SSL (capa de sockets seguros) para cifrar las comunicaciones. Si utiliza esta palabra clave, el servidor del repositorio debe estar configurado para SSL. Para obtener más información, consulte la documentación del administrador.
--server_url	-S	URL completa del servidor del repositorio. Utilice esta palabra clave si el URL del servidor incluye una raíz de contexto personalizada. Los valores de las palabras clave host, port y ssl no serán necesarios si especifica el URL del servidor.
--useDefault	-z	Utiliza la información de conexión del servidor definida en el archivo <code>Authorization.properties</code>
--help	-h	La información de ayuda del módulo de scripts

Conexiones de repositorio

Debe especificar el identificador de usuario y contraseña de Repositorio de IBM SPSS Collaboration and Deployment Services, así como la información del servidor del repositorio al final de cada comando.

Se pueden utilizar los métodos siguientes para proporcionar esta información de conexión:

- Utilización de palabras clave, como en los ejemplos siguientes:

```
--user usuario --password contraseña --host host --port puerto
--user usuario --password contraseña --server_url url
```

- Aunque el archivo `Authorization.properties`, donde el comando contiene un parámetro `--useDefault` (o la versión abreviada `-z`). Este método recupera la información de conexión del archivo `Authorization.properties`, que se encuentra en `carpeta de scripts\Lib\site-packages\config\Authorization.properties`. Utilice un editor de texto simple para modificar los siguientes valores en el archivo para igualar los ajustes de su repositorio:

```
# Información de autorización
user=admin
password=contraseña
host=su_host
port=80
```

O bien, puede utilizar la palabra clave `server_url` en el archivo de propiedades.

```
# Información de autorización
user=admin
password=contraseña
server_url=http://su_host:80/raíz_contexto
```

Los parámetros pasados por la línea de comandos siempre tienen prioridad. Por ejemplo, si se proporcionan `--user` y `--password` en la línea de comandos y también se proporciona el parámetro `--useDefault` o `-z`, se utilizan el usuario y la contraseña de la línea de comandos, y el host y el puerto se recuperan del archivo `Authorization.properties`. Como alternativa, si el usuario, contraseña, host y puerto se proporcionan en la línea de comandos, pero también se utiliza el parámetro `--useDefault` o el parámetro `-z`, `--useDefault` no se tiene en cuenta y sólo se utiliza la información de la línea de comandos.

Para todas las API descritas aquí, la sintaxis y ejemplos usan el parámetro `-z` para utilizar el número mínimo de parámetros obligatorios.

Scripts de repositorio de contenido

Los scripts del repositorio de contenido ofrecen la posibilidad de trabajar con recursos de repositorio, como archivos y carpetas. Esta área incluye las funciones siguientes:

- Creación y supresión de carpetas
- Carga y descarga de archivos
- Exportación e importación de carpetas
- Gestión de etiquetas, seguridad y metadatos

Esta sección describe el uso que la línea de comandos de Python hace de los procesos de funciones de repositorio. Cada función contiene información detallada sobre sintaxis, un ejemplo y los mensajes esperados.

Palabras clave

Tabla 3 enumera las palabras clave que admiten todas las funciones de repositorio. La segunda columna enumera cualquier versión opcional y abreviada de las palabras clave.

Importante: las palabras clave distinguen entre mayúsculas y minúsculas.

Tabla 3. Palabras clave de las API de repositorio.

Palabra clave	Versión abreviada opcional	Uso
--source	-s	Ruta de origen del archivo o carpeta
--target	-t	Ruta de la carpeta de destino
--version	-v	Versión de un archivo
--principal	-r	Usuario que necesita recibir permiso
--permission	-n	El tipo de permiso (como lectura, escritura, modificación y eliminación)
--label	-l	Etiqueta para asignar a una versión de un archivo
--criteria	-c	Criterios de búsqueda para encontrar los atributos de metadatos de archivos o carpetas
--author	-a	Nombre del autor de un archivo o carpeta
--description	-d	Descripción de un archivo o carpeta
--title	-i	Título de un archivo o carpeta
--expirationDate	-q	Fecha de caducidad de un archivo o carpeta
--expirationStartDate		Fecha de inicio de caducidad de un archivo o carpeta
--expirationEndDate		Fecha de finalización de caducidad de un archivo o carpeta
--keyword	-k	Palabra clave de un archivo o carpeta
--cascade	-x	Indica que los ajustes de seguridad de una carpeta deben propagarse a subcarpetas y archivos
--provider	-f	Proveedor de seguridad utilizado para recuperar los principales
--createVersion	-b	Indica que va a crearse una nueva versión de un archivo
--contentLanguage	-g	Idioma de contenido de un archivo o carpeta
--topic		Temas asignados a un archivo o carpeta. Puede introducir múltiples valores como --topic "tema1;tema2"
--modifiedBy		Usuario que ha modificado una archivo o carpeta

- *<fechaInicioCaducidad>* es la fecha de inicio de caducidad del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD (por ejemplo, 1997-07-16T19:20:30+01:00), donde:

AAAA = año en cuatro dígitos

MM = mes en dos dígitos (01 es enero, etc.)

DD = día del mes en dos dígitos (01 a 31)

hh = hora en dos dígitos (00 a 23, sin am/pm)

mm = minutos en dos dígitos (00 a 59)

ss = segundos en dos dígitos (00 a 59)

TZD = designador de zona horaria (Z o +hh:mm o -hh:mm)

- *<fechaFinCaducidad>* es la fecha de fin de caducidad del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaInicioCreación>* es la fecha de inicio de creación del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaFinCreación>* es la fecha de finalización de creación del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaIniModObjeto>* es la fecha de inicio de modificación del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaFinModObjeto>* es la fecha de finalización de modificación del archivo o carpeta. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaIniModVersión>* es la fecha de inicio de modificación de la versión. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- *<fechaFinModVersión>* es la fecha de finalización de modificación de la versión. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD.
- --submittedHierarchy indica que el archivo o carpeta se encuentra en la carpeta Submitted Jobs (trabajos enviados).

Todos los parámetros son opcionales.

Ejemplo

```
python CADSTool.py advanceSearch --label "Production" --keyword "Quarterly"
--useDefault -z
python CADSTool.py advanceSearch --createdStartDate "2009-12-01T00:00:00+01:00"
--createdEndDate "2010-12-15T21:33:40+01:00" -z
python CADSTool.py advanceSearch --uri "spsscr:///id=a010a37ba5992bb00000127b0f952f945be" -z
```

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Cuando la API se completa con éxito, aparece una lista de todos los archivos y carpetas que coinciden con los criterios de búsqueda. Esto suele incluir los nombres de archivo con su ruta completa y versiones.
- Error al buscar archivos y carpetas
- error: no existe tal opción:<opción>

La operación applySecurity

Establece la lista de control de acceso (ACL) de seguridad para un archivo o carpeta del repositorio.

Sintaxis

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

Donde:

- *<origen>* es la ruta completa de Repositorio de IBM SPSS Collaboration and Deployment Services del archivo o carpeta al que se aplicará la ACL de seguridad. Se trata de un parámetro obligatorio.
- *<principal>* es el usuario (como *admin*) que se aplicará al archivo o carpeta especificado como parte de la ACL. Se trata de un parámetro obligatorio.
- *<permiso>* es el tipo de permiso que se aplica al archivo o carpeta especificado (como lectura, escritura, modificación, supresión o propietario). Se trata de un parámetro obligatorio.
- *<proveedor>* es el proveedor de seguridad que se utilizará para recuperar información sobre los usuarios (principales). Se trata de un parámetro opcional. Los valores válidos son los siguientes:
 - Native para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado.
 - AD_*<nombre>* para Active Directory, donde *<nombre>* corresponde al nombre de proveedor de seguridad dentro del sistema
 - ADL_*<nombre>* para Active Directory con alteración temporal local, donde *<nombre>* corresponde al nombre del proveedor de seguridad dentro del sistema
 - ldap_*<nombre>* para OpenLDAP, donde *<nombre>* corresponde al nombre del proveedor de seguridad dentro del sistema
- --cascade se utiliza para definir la seguridad de una carpeta, para propagar los ajustes de seguridad a todos los archivos y subcarpetas en la carpeta especificada. Se trata de un parámetro opcional.

Ejemplos

El ejemplo siguiente aplica la seguridad a un archivo o carpeta:

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

El siguiente ejemplo aplica seguridad a una carpeta y todos sus archivos y subcarpetas:

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

El ejemplo siguiente aplica la seguridad a una carpeta para un principal en un proveedor de seguridad de Active Directory denominado SPSSAD:

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

El ejemplo siguiente aplica la seguridad a una carpeta para un principal en un proveedor de seguridad de OpenLDAP denominado LDAP:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Se ha definido correctamente el conjunto de permisos *<permiso>* para *<origen>*.
- *<origen>* El archivo o carpeta no existe. Vuelva a intentarlo.
- *<permiso>* Tipo de permiso no válido. Vuelva a intentarlo.
- *<origen>* Error al definir la ACL de seguridad.

La operación cascadeSecurity

Extiende los ajustes de seguridad de una carpeta a todos los archivos y subcarpetas de la carpeta.

Sintaxis

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

El valor de <origen> es la ruta completa de la carpeta en el repositorio. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- La ACL de seguridad se ha puesto en cascada correctamente para <origen>.
- <origen> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al representar en cascada la ACL de seguridad.

La operación copyResource

Copia un archivo o carpeta en otra carpeta del repositorio.

Se proporciona una función de cambio de nombre para esta API; donde es posible cambiar el nombre del archivo especificado cuando se copie. Los casos descritos al principio de “La operación moveResource” en la página 17 también se aplican a la API de copyResource.

Sintaxis

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

Donde:

- <origen> es la ruta completa en el repositorio de contenido de la carpeta o archivo que copiar. Se trata de un parámetro obligatorio.
- <destino> es la ruta completa en el repositorio de la carpeta o archivo que copiar. Se trata de un parámetro obligatorio.

Ejemplos

El siguiente ejemplo copia un archivo:

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

El siguiente ejemplo copia un archivo y cambia su nombre:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> se ha copiado correctamente.
- <origen> El archivo o carpeta no existe. Vuelva a intentarlo.
- <destino> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al copiar el archivo o carpeta.

La operación createFolder

Crea una nueva carpeta en una ubicación especificada del repositorio.

Sintaxis

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

El valor de *<origen>* es la ruta completa de la carpeta que se va a crear. Se trata de un parámetro obligatorio. En función en la ruta provista, se crea una nueva carpeta incluyendo cualquier subcarpeta.

Ejemplo

El ejemplo siguiente crea *Drafts* si aún no existe.

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- *<origen>* Carpeta creada correctamente.
- *<origen>* No existe esta carpeta. Vuelva a intentarlo.
- *<carpeta>* La carpeta ya existe. Vuelva a intentarlo.
- *<origen>* Error al crear la carpeta.

La operación deleteFile

Elimina un archivo del repositorio, incluyendo todas sus versiones.

Sintaxis

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

Donde:

- *<origen>* es la ruta completa en el repositorio del archivo que se quiere eliminar. Se trata de un parámetro obligatorio.
- *--submittedHierarchy* elimina un archivo de la carpeta Trabajos enviados. Se trata de un parámetro opcional.

Ejemplo

El siguiente ejemplo elimina un archivo del repositorio, incluyendo todas sus versiones:

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

El siguiente ejemplo elimina un archivo de la carpeta Trabajos enviados, incluyendo todas sus versiones:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- *<origen>* suprimido correctamente.
- *<origen>* No existe este archivo. Vuelva a intentarlo.
- *<origen>* Error al eliminar el archivo.

La operación deleteFileVersion

Elimina una versión específica de un archivo del repositorio.

Sintaxis

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

Donde:

- *<origen>* es la ruta completa en el repositorio del archivo que se quiere eliminar. Se trata de un parámetro obligatorio.
- *<versión>* es la versión específica del archivo que se desea eliminar. Se trata de un parámetro opcional.
- *<etiqueta>* es la etiqueta específica del archivo que se desea eliminar. Se trata de un parámetro opcional.
- `--submittedHierarchy` elimina una versión específica de un archivo de la carpeta Trabajos enviados. Se trata de un parámetro opcional.

Ejemplos

El siguiente ejemplo elimina una versión específica de un archivo:

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

El siguiente ejemplo elimina un archivo con una etiqueta específica:

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

El siguiente ejemplo elimina un archivo con una etiqueta específica de la carpeta Trabajos enviados:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- *<origen>* suprimido correctamente.
- *<origen>* No existe este archivo. Vuelva a intentarlo.
- *<origen>* Error al eliminar el archivo.

La operación deleteFolder

Suprime una carpeta del repositorio, incluyendo todo su contenido.

Sintaxis

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

Donde:

- *<origen>* es la ruta completa en el repositorio de la carpeta que se quiere eliminar. Se trata de un parámetro obligatorio.
- `--submittedHierarchy` elimina una versión específica de una carpeta dentro de la carpeta Trabajos enviados. Se trata de un parámetro opcional.

Ejemplos

El siguiente ejemplo elimina una carpeta:

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

El siguiente ejemplo elimina una carpeta de la carpeta Trabajos enviados:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- *<origen>* suprimido correctamente.

- <origen> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al eliminar la carpeta.

La operación downloadFile

Descarga la versión específica de un archivo del repositorio al sistema de archivos local.

Sintaxis

```
python CADSTool.py downloadFile --source "<origen>" --version "<versión>" --label "<etiqueta>" --target "<destino>" -z
```

Donde:

- <origen> es la ruta completa de repositorio. Se trata de un parámetro obligatorio.
- <versión> es la versión del archivo que se desea descargar. Se trata de un parámetro opcional.
- <etiqueta> es la etiqueta del archivo que se desea descargar. Se trata de un parámetro opcional.
- <destino> es la ruta completa (en el sistema de archivos local) donde va a descargarse el archivo.

Ejemplos

El siguiente ejemplo descarga la última versión de un archivo:

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

El siguiente ejemplo descarga una versión específica de un archivo usando un marcador de versión:

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

El siguiente ejemplo descarga una versión etiquetada de un archivo:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> Archivo descargado correctamente.
- <origen> No existe este archivo. Vuelva a intentarlo.
- <destino> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al descargar el archivo.

La operación export

Inicia una exportación desde el repositorio de contenido, permitiendo al usuario que seleccione qué archivos y carpetas exportar y guardando el archivo de exportación *.pes en el sistema de archivos local.

Sintaxis

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

Donde:

- <origen> es la ruta completa en el repositorio de la carpeta que se quiere exportar. Se trata de un parámetro obligatorio.
- <destino> es la ruta completa (en el sistema de archivos local) para el archivo de exportación *.pes que se va a crear. Se trata de un parámetro obligatorio.

Ejemplo

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


Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> exportado correctamente.
- <origen> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al exportar la carpeta.

La operación `getAccessControlList`

Define la lista de control de acceso de seguridad (ACL) para un archivo o carpeta especificado en el repositorio de contenido.

Sintaxis

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

El valor de <origen> es la ruta completa del archivo o carpeta. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> El archivo o carpeta no existe. Vuelva a intentarlo.
- Error al recuperar los detalles de seguridad de <origen>.

La operación `getAllVersions`

Recupera una lista de todas las versiones de un archivo en el repositorio.

Sintaxis

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

Donde:

- <origen> es la ruta completa en el repositorio del archivo cuyas versiones se quieren recuperar. Se trata de un parámetro obligatorio.
- --submittedHierarchy recupera versiones de la carpeta Trabajos enviados. Se trata de un parámetro opcional.

Ejemplos

El siguiente ejemplo recupera todas las versiones de un archivo especificado:

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

El siguiente ejemplo recupera todas las versiones de un archivo especificado de la carpeta Trabajos enviados:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> No existe este archivo. Vuelva a intentarlo.
- <origen> Error al recuperar las versiones del archivo.
- Cuando el proceso se complete con éxito, se mostrará la información sobre cada versión de archivo, incluyendo información sobre la etiqueta y el marcador de versión.

La operación getChildren

Recupera una lista de todos los archivos y carpetas de una carpeta especificada del repositorio.

Sintaxis

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

El valor de <origen> es la ruta completa de la carpeta. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Cuando el comando se completa con éxito, enumera todo el contenido de la carpeta especificada.
- <origen> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al obtener los recursos.

La operación getCustomPropertyValue

Recupera los valores válidos aceptados por una propiedad personalizada especificada.

Sintaxis

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

El valor de <nombrePropiedad> es el nombre de la propiedad personalizada. Se trata de un parámetro opcional.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <nombrePropiedad> toma valores como <valores válidos>
- Error al recuperar los detalles de propiedad para <nombrePropiedad>.

La operación getMetadata

Recupera los atributos de metadatos de un archivo o carpeta en el repositorio.

Sintaxis

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

Donde:

- <origen> es la ruta completa de repositorio del archivo o carpeta para el que se van a recuperar los metadatos. En las carpetas se ignoran los atributos de versión y etiqueta. Se trata de un parámetro obligatorio.
- <versión> es la versión del archivo para el que se van a recuperar los metadatos. Se trata de un parámetro opcional.
- <etiqueta> es la etiqueta del archivo para el que se van a recuperar los metadatos. Se trata de un parámetro opcional.
- --submittedHierarchy recupera metadatos de la carpeta Trabajos enviados. Se trata de un parámetro opcional.

Ejemplos

El ejemplo siguiente recupera los metadatos de una carpeta:

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

El ejemplo siguiente recupera los metadatos de una versión etiquetada de un archivo:

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

El siguiente ejemplo recupera los metadatos de una versión etiquetada de un archivo dentro de la carpeta Submitted Jobs (Trabajos enviados):

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- `<origen>` No existe este archivo. Vuelva a intentarlo.
- `<origen>` Error al recuperar los metadatos del archivo.
- Cuando el proceso se complete con éxito, se mostrará la información de metadatos del archivo o carpeta, incluyendo las propiedades de metadatos personalizados.

La operación de importación

Importa un archivo de exportación *.pes existente del sistema de archivo local en el repositorio.

Sintaxis

```
python CADSTool.py import --source "<origen>" --target "<destino>"  
--resourceType "<tipo>"  
--resourceConflict "<conflicto_rec>"  
--labelFrom "<etiqueta>"  
--lockResolution "<resolución>"  
--invalidVersionConflict "<conflicto_ver>"  
--resourceDef "<definición_rec>"  
--exclude "<exclusión>"  
-z
```

Donde:

- `<origen>` es la ruta completa (en el sistema de archivos local) del archivo de exportación *.pes que se va a importar en el repositorio. Se trata de un parámetro obligatorio.
- `<destino>` es la ruta completa del repositorio en el que se va a importar el archivo de exportación *.pes. Se trata de un parámetro obligatorio.
- `<tipo>` indica el tipo de contenido que se va a importar. Especifique uno de los valores siguientes:
 - **ContentRepository** para objetos de contenido como archivos y carpetas
 - **ResourceDef** para definiciones de recursos
 - **Credential** para credenciales de usuario
 - **DataSource** para definiciones de orígenes de datos
 - **MessageDomain** para dominios de mensajes
 - **ServerCluster** para definiciones de clúster de servidores
 - **Server** para definiciones de servidor
 - **PromotionPolicy** para políticas de promoción

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **ContentRepository**.

- `<conflicto_rec>` indica cómo resolver conflictos de ID o nombres duplicados. Especifique uno de los valores siguientes:

- **keepTarget**. Se mantendrá el elemento de destino. El elemento de origen con el ID duplicado, contenido en el archivo *.pes*, no se tendrá en cuenta.
- **addNewVersion**. Esta opción se suele utilizar para resolver conflictos de ID o conflictos de nombre. Si se produce un conflicto de ID duplicado entre el objeto de origen y de destino, se creará una nueva versión del objeto en la ubicación de destino. Si se produce un conflicto de nombre, se cambiará el nombre del objeto importado en la ubicación de destino. Normalmente, a los objetos cuyos nombres se cambian se les añade *_1*, *_2* y así sucesivamente. En el caso de que un evento con dos versiones de objeto tengan la misma etiqueta, el sistema mantiene una etiqueta y desecha la etiqueta duplicada porque dos versiones del mismo elemento no pueden tener la misma etiqueta. La etiqueta que se conserva depende del parámetro **labelFrom**.

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **keepTarget**.

- *<etiqueta>* especifica qué etiqueta se utiliza si dos versiones de un objeto tienen la misma etiqueta. Se descarta la etiqueta de la otra versión. Especifique uno de los valores siguientes:
 - **origen**
 - **destino**

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **source**.

- *<resolución>* define cómo actuar si se encuentran recursos bloqueados. Especifique uno de los valores siguientes:
 - **continue**. La importación continúa, omitiendo los recursos bloqueados.
 - **abort**. Termina el proceso de importación si se encuentran recursos bloqueados. Si se encuentran conflictos debidos a bloqueos de objetos, el proceso de importación terminará y fallará.

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **abort**.

- *<conflicto_ver>* define cómo proceder si se encuentra una versión no válida durante el proceso de importación. Especifique uno de los valores siguientes:
 - **import**. Se importará la versión válida.
 - **discard**. La versión no válida se eliminará.

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **import**.

- *<definición_rec>* define el comportamiento de proceso para definiciones de recursos. Especifique uno de los valores siguientes:
 - **recommended**. Una definición de recurso sólo se importa si el identificador o nombre no entra en conflicto con una definición de destino. Las definiciones de recursos que tengan algún conflicto no se importan.
 - **include**. Se importan todas las definiciones de recursos del archivo de importación. Puede seleccionar uno o más tipos de definiciones de recursos para excluirlos de la importación marcando el recuadro de selección correspondiente.
 - **exclude**. No se importan definiciones de recursos desde los archivos de importación. Los objetos importados podrían necesitar modificación para hacer referencia a definiciones de recursos disponibles.

Se trata de un parámetro opcional. Si no se especifica este parámetro, se utilizará el valor predeterminado de **recommended**.

- *<exclusión>* define qué tipos de recursos se excluyen durante la importación. Se pueden combinar varios valores en cualquier orden como una lista delimitada por el carácter de punto y coma. Especifique uno o más de los valores siguientes:
 - **credential** excluye credenciales de usuario
 - **customproperty** excluye propiedades personalizadas para objetos de recursos

- **datasource** excluye definiciones de orígenes de datos
- **messagedomain** excluye dominios de mensajes
- **notification** excluye definiciones de notificaciones
- **servercluster** excluye definiciones de clúster de servidores
- **server** excluye definiciones de servidor
- **topic** excluye definiciones de tema

Se trata de un parámetro opcional. Si no se especifica este parámetro, se incluyen todos los tipos en la importación.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> importado correctamente.
- <origen> No existe este archivo. Vuelva a intentarlo.
- <destino> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al importar la carpeta.

La operación moveResource

Mueve un archivo o carpeta a otra carpeta del repositorio.

Se proporciona una función de cambio de nombre para esta API; donde es posible cambiar el nombre del archivo o carpeta especificado cuando se mueva. Los siguientes casos describen el comportamiento de la función de cambio de nombre:

Si el origen es */Temp Folder/Temp.txt* y el destino es */Carpeta Demo*:

- **Caso 1:** si la carpeta *Demo Folder* existe, *Temp.txt* se mueve a *Demo Folder*.
- **Caso 2:** si la carpeta *Demo Folder* no existe, *Temp.txt* se mueve a */* y se cambia su nombre a *Demo Folder*.

Si el origen es */Temp Folder/Temp.txt* y el destino es */Carpeta Demo/Abc.dat*:

- **Caso 1:** si la carpeta *Demo Folder* existe, *Temp.txt* se mueve a *Demo Folder* y se cambia su nombre a *Abc.dat*.
- **Caso 2:** si la carpeta *Demo Folder* no existe, aparece un error.

Sintaxis

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

Donde:

- <origen> es la ruta completa en el repositorio de la carpeta o archivo que se va a mover. Se trata de un parámetro obligatorio.
- <destino> es la ruta completa en el repositorio donde se va a mover la carpeta o archivo. Se trata de un parámetro obligatorio.

Ejemplos

El siguiente ejemplo mueve un archivo:

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

El siguiente ejemplo mueve una carpeta:

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

El siguiente ejemplo mueve un archivo y cambia su nombre:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- `<origen>` se ha movido correctamente.
- `<origen>` El archivo o carpeta no existe. Vuelva a intentarlo.
- `<destino>` No existe esta carpeta. Vuelva a intentarlo.
- `<origen>` Error al mover el archivo o carpeta.

La operación `removeLabel`

Quita una etiqueta de un archivo del repositorio.

Sintaxis

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

Donde:

- `<origen>` es la ruta completa del archivo en el repositorio. Se trata de un parámetro obligatorio.
- `<etiqueta>` es el nombre de etiqueta que se quitará del archivo especificado. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Etiqueta eliminada correctamente para `<origen>`.
- `<origen>` No existe esta carpeta. Vuelva a intentarlo.
- `<origen>` Error al suprimir la etiqueta.
- `<etiqueta>` No existe esta etiqueta. Vuelva a intentarlo.

La operación `removeSecurity`

Define la lista de control de acceso de seguridad (ACL) para un archivo o carpeta especificado del repositorio.

Sintaxis

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

Donde:

- `<origen>` es la ruta completa de la carpeta o archivo cuya seguridad se va a eliminar. Se trata de un parámetro obligatorio.
- `<principal>` es el usuario/principal (como *admin*) para el que se quitará la seguridad del archivo o carpeta especificado. Se trata de un parámetro obligatorio.
- `<proveedor>` es el proveedor de seguridad que se utilizará para recuperar información sobre los usuarios (principales). Se trata de un parámetro opcional. Los valores válidos son los siguientes:

- Native para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado.
- AD_<nombre> para Active Directory, donde <nombre> corresponde al nombre de proveedor de seguridad dentro del sistema
- ADL_<nombre> para Active Directory con alteración temporal local, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema
- ldap_<nombre> para OpenLDAP, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema
- --cascade se utiliza al quitar la seguridad de una carpeta, para quitar los ajustes de seguridad de todos los archivos y subcarpetas de la carpeta especificada. Se trata de un parámetro opcional.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> Todas las ACL de seguridad se han eliminado correctamente.
- <origen> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al eliminar la ACL de seguridad.

La operación search

Busca archivos y carpetas en el repositorio. Los resultados son una lista de archivos o carpetas que coinciden con los criterios de búsqueda, junto con sus versiones.

Sintaxis

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

El valor de <critérios> es la cadena de búsqueda utilizada para buscar los metadatos en todos los archivos y carpetas del repositorio. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Cuando la búsqueda se completa con éxito, aparece una lista de todos los archivos y carpetas que coinciden con los criterios de búsqueda. Esto suele incluir los nombres de archivo con su ruta completa y versiones.
- <critérios> Ningún archivo o carpeta cumple los criterios de búsqueda.
- Error al buscar archivos y carpetas.

La operación setLabel

Aplica una etiqueta a una versión de un archivo en el repositorio. Si el archivo ya se ha etiquetado, la etiqueta original se quita y se reemplaza por una nueva.

Sintaxis

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

Donde:

- <origen> es la ruta completa del archivo en el repositorio. Se trata de un parámetro obligatorio.

- `<versión>` es la versión del archivo que se desea descargar. Se trata de un parámetro obligatorio.
- `<etiqueta>` es el nombre de etiqueta que se aplicará a la versión especificada del archivo. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Etiqueta establecida correctamente para `<origen>`.
- `<origen>` No existe esta carpeta. Vuelva a intentarlo.
- `<origen>` Error al definir la etiqueta.

La operación setMetadata

Aplica las propiedades de metadatos a los archivos y carpetas del repositorio.

Tabla 4 muestra las propiedades de metadatos y si se pueden aplicar a archivos y carpetas.

Tabla 4. Propiedades de metadatos y tipos de recursos.

Propiedad de metadatos	Tipo de recurso
Autor	Archivo
Descripción	Archivo o carpeta
Título	Archivo o carpeta
Fecha caducidad	Archivo o carpeta
Palabra clave	Archivo
Temas	Archivo
Metadatos personalizados	Archivo o carpeta

Sintaxis

```
python CADSTool.py setMetadata --source "<origen>" --version "<versión>" --label "<etiqueta>" --author "<autor>" --title "<título>" --description "<descripción>" --expirationDate "<fechaCaducidad>" --topic "<tema>" --keyword "<palabraClave>" --customProperty "<propiedadPersonalizada>" -z
```

Donde:

- `<origen>` es la ruta completa en el repositorio del archivo o carpeta cuyos metadatos se quieren recuperar. Se trata de un parámetro obligatorio.
- `<autor>` es el autor del archivo o carpeta. Se trata de un parámetro opcional.
- `<título>` es el título del archivo o carpeta. Se trata de un parámetro opcional.
- `<descripción>` es la descripción del archivo o carpeta. Se trata de un parámetro opcional.
- `<fechaCaducidad>` es la fecha de caducidad del archivo o carpeta. Se trata de un parámetro opcional. El formato de fecha es YYYY-MM-DDThh:mm:ssTZD (por ejemplo, 1997-07-16T19:20:30+01:00), donde:

AAAA = año en cuatro dígitos

MM = mes en dos dígitos (01 es enero, etc.)

DD = día del mes en dos dígitos (01 a 31)

hh = hora en dos dígitos (00 a 23, sin am/pm)

mm = minutos en dos dígitos (00 a 59)

ss = segundos en dos dígitos (00 a 59)

TZD = designador de zona horaria (Z o +hh:mm o -hh:mm)

- *<palabraClave>* es la palabra clave del archivo o carpeta. Se trata de un parámetro opcional.
- *<versión>* es la versión específica del archivo a la que aplicar los metadatos. Se trata de un parámetro opcional.
- *<etiqueta>* es la versión etiquetada del archivo a la que aplicar los metadatos. Se trata de un parámetro opcional.
- *<tema>* es el tema que se va a aplicar al archivo o carpeta. Se trata de un parámetro opcional.
- *<propiedadPersonalizada>* es la propiedad personalizada que se va a aplicar al archivo o carpeta. Se trata de un parámetro opcional. Los valores se especifican como *<propiedadPersonalizada>=<valor>*. Para aplicar más de una propiedad personalizada, utilice punto y coma (;) para separarlas (*<customProperty>=<valor>;<customProperty>=<valor>*). Si tiene valores de propiedad de selección múltiple, sepárelos con el operador | (*<customProperty>=opt1|opt2;<customProperty>=value*).

Nota: Es necesario indicar al menos un parámetro opcional para usar la API setMetadata.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- *<origen>* Metadatos definidos correctamente.
- *<origen>* El archivo o carpeta no existe. Vuelva a intentarlo.
- *<origen>* Error al definir los metadatos.

La operación uploadFile

Guarda un archivo en el repositorio de contenido del sistema de archivos local, con la opción de crear una nueva versión del archivo si éste ya existe.

Sintaxis

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

Donde:

- *<origen>* es la ruta completa (en el sistema de archivos local) del archivo que se desea cargar. Se trata de un parámetro obligatorio.
- *<destino>* es la ruta completa de la carpeta en el repositorio donde se va a cargar el archivo. Se trata de un parámetro obligatorio.
- *--createVersion* indica que el archivo especificado ya sale y que es necesario crear una nueva versión. Se trata de un parámetro opcional.

Ejemplos

En el ejemplo siguiente, el destino es una ruta completa de *Drafts*:

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

Si *MyReport.rptdesign* ya existe en la carpeta */Demo/Drafts*, utilice el parámetro *--createVersion*:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> Archivo cargado correctamente.
- <origen> No existe este archivo. Vuelva a intentarlo.
- <destino> No existe esta carpeta. Vuelva a intentarlo.
- <origen> Error al cargar el archivo.

Funciones de gestión de procesos

Los scripts de gestión de procesos ofrecen la posibilidad de trabajar con trabajos. Esta área incluye las funciones siguientes:

- Ejecución de trabajos
- Recuperación de historiales de trabajos
- Recuperación de detalles de trabajos

Esta sección describe el uso que la línea de comandos de Python hace de los procesos de funciones de gestión de procesos. Cada API contiene información detallada sobre sintaxis, un ejemplo y los mensajes esperados.

Palabras clave

Tabla 5 muestra las palabras clave admitidas por las API de gestión de procesos. La segunda columna enumera cualquier versión opcional y abreviada de las palabras clave. La tabla sólo enumera palabras clave específicas para las API de gestión de procesos. Para obtener palabras clave adicionales que se pueden aplicar tanto a las API de gestión de procesos como a las API de repositorio, consulte Tabla 2 en la página 3 y Tabla 3 en la página 5.

Tabla 5. Palabras clave de las API de gestión de procesos

Palabra clave	Versión abreviada opcional	Uso
--source	-s	El nombre completo del archivo, incluyendo la ruta.
--target	-t	Ruta de la carpeta de destino
--notification	-j	Indica que el trabajo se ejecutará con notificaciones
--async	-m	Indica que el trabajo se ejecutará de forma asíncrona
--execId	-y	ID de ejecución del trabajo
--jobStepName	-q	Nombre del paso de trabajo.
--log		Indica que no se eliminarán los registros. Si se usa junto con --target, los registros se almacenarán en una ubicación especificada por --target. De lo contrario, los registros se mostrarán en línea.

Operaciones de gestión de procesos

La operación deleteJobExecutions

Suprime los objetos de ejecución de trabajos especificados.

Sintaxis

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

El valor de <IDejec> es una lista delimitada por espacios de identificadores de las ejecuciones a suprimir. Se trata de un parámetro obligatorio.

Ejemplos

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- ID de ejecución no especificado.

La operación executeJob

Ejecuta un trabajo de forma sincrónica o asíncrona en función de los parámetros pasados. En el caso de una ejecución asíncrona, la API no devuelve el control hasta que se completa el trabajo. En el caso de una ejecución asíncrona, la API devuelve el control después de que se inicie el trabajo.

Sintaxis

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

Donde:

- <origen> es la ruta completa del trabajo en el repositorio. Se trata de un parámetro obligatorio.
- --notification se utiliza para ejecutar el trabajo con notificaciones. Se trata de un parámetro opcional.
- --async se utiliza para ejecutar el trabajo de forma asíncrona. Se trata de un parámetro opcional.

Ejemplos

El siguiente ejemplo ejecuta el trabajo de forma sincrónica sin notificaciones:

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

El siguiente ejemplo ejecuta el trabajo de forma sincrónica con notificaciones:

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

El siguiente ejemplo ejecuta el trabajo de forma asíncrona sin notificaciones:

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

El siguiente ejemplo ejecuta el trabajo de forma asíncrona con notificaciones:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- <origen> Trabajo ejecutado correctamente. El ID de ejecución de trabajo es <execId>.
- <origen> No existe este trabajo. Vuelva a intentarlo.
- <origen> Error al recuperar el trabajo.

La operación getJobExecutionDetails

Enumera los detalles de ejecución de un trabajo específico, incluyendo cualquier paso e iteraciones del trabajo.

Sintaxis

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

Donde:

- <IDejec> es el identificador de ejecución del trabajo. Se trata de un parámetro obligatorio.
- --log indica que el registro del trabajo debe mostrarse en línea. Si no se incluye el parámetro --log, cualquier registro generado por una ejecución de paso de trabajo no se mostrará. Se trata de un parámetro opcional.
- <destino> es la ubicación (en el sistema de archivos local) para almacenar los registros. Se trata de un parámetro opcional, y sólo se usa en conjunto con el parámetro --log.

Ejemplos

El siguiente ejemplo enumera los detalles de una ejecución de trabajo específica:

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

El siguiente ejemplo enumera los detalles de una ejecución de trabajo específica con el registro mostrado en línea:

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

El siguiente ejemplo enumera los detalles de una ejecución de trabajo específica con los registros del trabajo almacenados en una ubicación específica:

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- Para que la ejecución se realice con éxito, todos los detalles de la ejecución se enumeran para el trabajo, los pasos del trabajo y las iteraciones del trabajo. Los registros se muestran en línea o se guardan en una ubicación especificada en el sistema de archivos local.
- <IDejec> No existe esta ejecución. Vuelva a intentarlo.
- <IDejec> Error al mostrar los detalles de una ejecución de trabajo.
- --target no puede usarse sin el parámetro --log

La operación getJobExecutionList

Enumera las ejecuciones actuales y completadas de un trabajo específico para todas las versiones del trabajo.

Sintaxis

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

El valor de <origen> es la ruta completa del trabajo en el repositorio. Se trata de un parámetro obligatorio.

Ejemplo

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

Mensajes

Los siguientes mensajes pueden aparecer al utilizar esta API:

- En una ejecución satisfactoria del trabajo especificado, se enumeran todos los detalles de la ejecución como el ID de ejecución, el nombre del trabajo, el estado de ejecución del trabajo y la hora de inicio y fin de la ejecución.
- <origen> No existe este trabajo. Vuelva a intentarlo.
- <origen> Error al mostrar la lista de ejecución de un trabajo.

Capítulo 3. El módulo PESImpl

El recurso IBM SPSS Collaboration and Deployment Services - Essentials for Python permite la interacción con objetos del Repositorio de IBM SPSS Collaboration and Deployment Services directamente mediante procesos en Python.

En su código Python, importe la clase PESImpl desde el módulo `pes.api.PESImpl`. Cree un objeto PESImpl que utilice la información de conexión del repositorio al que vaya a conectarse.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("usuario", "contraseña", "host", "puerto", ssl=True)
```

Los parámetros del constructor de PESImpl son los siguientes:

- *usuario* corresponde al nombre de usuario. El valor debe incluir un prefijo que indica el proveedor de seguridad si el usuario no procede del proveedor nativo. Son válidos los valores de prefijo siguientes:
 - Native para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado.
 - AD_<nombre> para Active Directory, donde <nombre> corresponde al nombre de proveedor de seguridad dentro del sistema
 - ADL_<nombre> para Active Directory con alteración temporal local, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema
 - ldap_<nombre> para OpenLDAP, donde <nombre> corresponde al nombre del proveedor de seguridad dentro del sistema

Tras el prefijo, coloque una barra inclinada y el nombre de usuario. Para proveedores de Active Directory, incluya el dominio en el prefijo. Por ejemplo, para el usuario *icrod* del dominio *MYDOMAIN* de la instancia de Active Directory *AD_SPSSAD*, el valor es *AD_SPSSAD/MYDOMAIN/icrod*. Si el usuario *icrod* está en el proveedor de OpenLDAP *SPSSLDAP*, el valor será *ldap_SPSSLDAP/icrod*.

- *contraseña* corresponde a la contraseña asociada con el usuario especificado
- *host* designa el nombre del servidor del repositorio
- *puerto* especifica el número de puerto del servidor del repositorio
- *ssl=True* indica que el servidor del repositorio utiliza el protocolo SSL (capa de sockets seguros) para cifrar las comunicaciones. Si el parámetro *ssl* se establece en *False*, o si el parámetro se omite al crear el objeto PESImpl, las comunicaciones del servidor no utilizarán SSL. Cuando utilice SSL, el servidor del repositorio debe estar configurado para SSL. Para obtener más información, consulte la documentación del administrador.

Como alternativa, puede especificar el URL del servidor en lugar de los parámetros *host*, *puerto* y *ssl*.

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("usuario", "contraseña", server_url="url")
```

El parámetro *server_url* especifica el URL completo del servidor del repositorio. Utilice este parámetro si el servidor utiliza una ruta de contexto personalizada. Por ejemplo, el constructor siguiente corresponde a un servidor denominado *miServidor* que utiliza SSL en el puerto 443 y que tiene una ruta de contexto de */ibm/spss*:

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("miUsuario", "miContraseña", server_url="https://miServidor:443/ibm/spss")
```

Nota: Una dirección IPv6 entre corchetes como, por ejemplo, `[3ffe:2a00:100:7031::1]`.

Es posible acceder después a métodos específicos utilizando el objeto `pesImpl`.

API del repositorio de contenido

Los scripts del repositorio de contenido ofrecen la posibilidad de trabajar con recursos de repositorio, como archivos y carpetas. Esta área incluye las funciones siguientes:

- Creación y supresión de carpetas
- Carga y descarga de archivos
- Exportación e importación de carpetas
- Gestión de etiquetas, seguridad y metadatos

En esta sección se describe la API PESImpl utilizada para trabajar con recursos almacenados en el repositorio. Cada método contiene información detallada sobre sintaxis, un ejemplo y los mensajes esperados.

Métodos

Las secciones siguientes muestran todos los métodos de repositorio de contenido compatibles con IBM SPSS Collaboration and Deployment Services.

Nota:

- Para todos los métodos con parámetros opcionales `Label` y `Version`, utilice `Label` o `Version`, pero no ambos. Si no se especifica `Version` ni `Label` para un archivo o carpeta determinado, se utiliza la versión más reciente.
- Para todos los métodos que requieren una ruta a los archivos o carpetas del repositorio, puede utilizarse la ruta o el URI del objeto. El URI de objeto puede obtenerse visualizando las propiedades de objeto en IBM SPSS Collaboration and Deployment Services Deployment Manager.
- En el caso de métodos que requieran la introducción de un repositorio de origen o destino o de rutas de sistema de archivos que contengan caracteres Unicode no latinos, las cadenas deben especificarse como objetos Unicode, por ejemplo:

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

El método `advanceSearch`

Busca archivos y carpetas en el repositorio en función de varios parámetros pasados como entrada.

Puede buscar los siguientes elementos:

- Autor
- Descripción
- Título
- Creado por
- Modificado por
- Fecha de caducidad inicial
- Fecha de caducidad final
- Tipo MIME
- Etiqueta
- Palabra clave
- Temas
- Fecha de inicio de creación
- Fecha de finalización de creación
- Fecha de inicio de modificación de versión
- Fecha de finalización de modificación de objeto

- Fecha de inicio de modificación de objeto
- Fecha de inicio de modificación de versión
- Fecha de finalización de modificación de versión
- URI de carpeta padre
- URI del recurso

`advanceSearch(criteriaDict,submittedHierarchy)`

Tabla 6. Parámetros de entrada de `advanceSearch`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>criteriaDict</code>	Obligatorio	Diccionario	<p>El diccionario contiene la clave/valor del par con el que se realizará la búsqueda. Los valores de clave aceptables son:</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>
<code>submittedHierarchy</code>	Opcional	Booleano	Indica si se debe buscar en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tenga en cuenta que actualmente `expirationStartDate` y `expirationEndDate` no funcionan si se usan junto con otros campos de búsqueda (como título o autor).

Tabla 7. Valor de retorno de `advanceSearch`.

Tipo	Descripción
PageResult	Estructura en la que cada fila corresponde a una coincidencia de búsqueda. Consulte el tema “La clase PageResult” en la página 55 para obtener más información.

Tabla 8. Excepciones de `advanceSearch`.

Tipo	Descripción
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo: búsqueda por etiqueta y palabra clave

El ejemplo siguiente devuelve todas las versiones de archivo etiquetadas con *Production* (producción) que tienen un valor de palabra clave de *Quarterly* (trimestralmente).

```
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()
```

Ejemplo: búsqueda por URI

El ejemplo siguiente devuelve todas las versiones de archivo del archivo que tiene el URI especificado.

```
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()
```

El método `applySecurity`

Establece la lista de control de acceso (ACL) de seguridad para un archivo o carpeta del repositorio.

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

Tabla 9. Parámetros de entrada de `applySecurity`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000 010dcee0eaa28219"

Tabla 9. Parámetros de entrada de *applySecurity* (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>principal</i>	Obligatorio	Cadena	El usuario (por ejemplo <i>admin</i>) que se aplica al archivo o carpeta especificado como parte de la ACL	administrador
<i>permission</i>	Obligatorio	Cadena	El tipo de permiso que se aplicará al archivo o carpeta especificado	READ, WRITE, DELETE, MODIFY_ACL, OR OWNER
<i>provider</i>	Opcional	Cadena	<p>Proveedor de seguridad que se usará para aplicar seguridad a los usuarios (como <i>Native</i>). Los valores válidos son los siguientes:</p> <ul style="list-style-type: none"> • <i>Native</i> para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado. • <i>AD_<nombre></i> para Active Directory, donde <i><nombre></i> corresponde al nombre de proveedor de seguridad dentro del sistema • <i>ADL_<nombre></i> para Active Directory con alteración temporal local, donde <i><nombre></i> corresponde al nombre del proveedor de seguridad dentro del sistema • <i>ldap_<nombre></i> para OpenLDAP, donde <i><nombre></i> corresponde al nombre del proveedor de seguridad dentro del sistema 	<i>Native</i>
<i>cascade</i>	Opcional	Booleano	Extiende los ajustes de seguridad a todos los archivos y subcarpetas de la carpeta especificada	True o False

Tabla 10. Valor de retorno de *applySecurity*.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 11. Excepciones de *applySecurity*.

Tipo	Descripción
<i>ResourceNotFoundException</i>	El archivo de origen no existe.
<i>InsufficientParameterException</i>	No se han especificado parámetros obligatorios.
<i>IllegalParameterException</i>	El usuario o nombre de proveedor de seguridad especificado es incorrecto.

Ejemplo

El ejemplo siguiente asigna el permiso *READ* sobre el archivo designado a un usuario.

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

El método `cascadeSecurity`

Extiende los ajustes de seguridad de una carpeta a todos los archivos y subcarpetas de la carpeta.

```
cascadeSecurity(source)
```

Tabla 12. Parámetros de entrada de `cascadeSecurity`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto de la carpeta en el repositorio	"/Temp Folder" o "0a58c3670016a7860000 010dcee0eaa28219"

Tabla 13. Valor de retorno de `cascadeSecurity`.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 14. Excepciones de `cascadeSecurity`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	La carpeta de origen no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente organiza en cascada la seguridad de la carpeta *Projects* con el contenido de la carpeta.

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

El método `copyResource`

Copia un archivo o carpeta en otra carpeta del repositorio. Es posible cambiar el nombre del archivo o carpeta especificado cuando se copie.

Consulte “El método `moveResource`” en la página 46 para obtener más información sobre cambios de nombre.

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

Tabla 15. Parámetros de entrada de `copyResource`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000 010dcee0eaa28219"

Tabla 15. Parámetros de entrada de *copyResource* (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>target</i>	Obligatorio	Cadena	Ruta completa o URI de objeto de la carpeta a la que copiar el archivo. También es posible indicar un nuevo nombre de archivo para cambiar el nombre del archivo o la carpeta especificada cuando se copia.	"/New Folder" o "/New Folder/abc.dat"

Tabla 16. Valor de retorno de *copyResource*.

Tipo	Descripción
String	URI del archivo o carpeta que se ha copiado

Tabla 17. Excepciones de *copyResource*.

Tipo	Descripción
ResourceNotFoundException	El archivo de origen o la carpeta de destino no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente copia la carpeta *Drafts* a una carpeta denominada *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
```

El método createFolder

Creará una nueva carpeta en una ubicación especificada del repositorio.

```
createFolder(source)
```

Tabla 18. Parámetros de entrada de *createFolder*.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	La carpeta a crear en el repositorio	/Temp Folder/Sample Folder

Tabla 19. Valor de retorno de *createFolder*.

Tipo	Descripción
String	URI de la carpeta creada

Tabla 20. Excepciones de *createFolder*.

Tipo	Descripción
InsufficientParameterException	No se han especificado parámetros obligatorios.
ResourceAlreadyExistsException	La carpeta especificada ya existe en el repositorio.

Ejemplo

El ejemplo siguiente crea una carpeta denominada *Drafts* como un hijo de la carpeta *Demo*. Si se produce un problema al crear la carpeta, se envía un mensaje de excepción a la consola.

```

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."

```

El método deleteFile

Suprime un archivo del repositorio. Se suprimen todas las versiones del archivo.

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

Tabla 21. Parámetros de entrada de deleteFile.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000010dcee0eaa28219"
<i>submittedHierarchy</i>	Opcional	Booleano	Indica si el archivo se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 22. Valor de retorno de deleteFile.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 23. Excepciones de deleteFile.

Tipo	Descripción
ResourceNotFoundException	El archivo de origen no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.
IllegalParameterException	El recurso especificado a suprimir es una carpeta.

Ejemplo

El ejemplo siguiente suprime el archivo *MyReport.rptdesign* del repositorio.

```

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."

```

El método deleteFileVersion

Elimina una versión específica de un archivo del repositorio.

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

Tabla 24. Parámetros de entrada de deleteFileVersion.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000010dcee0eaa28219"
<i>version</i>	Opcional. No obstante, debe especificarse <i>version</i> o <i>label</i> .	Cadena	Versión específica del archivo que se desea eliminar	"0:2006-08-25 21:15:49.453"
<i>label</i>	Opcional. No obstante, debe especificarse <i>version</i> o <i>label</i> .	Cadena	Versión etiquetada específica del archivo que se desea eliminar	"Version 1"
<i>submittedHierarchy</i>	Opcional	Booleano	Indica si el archivo se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 25. Valor de retorno de deleteFileVersion.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 26. Excepciones de deleteFileVersion.

Tipo	Descripción
ResourceNotFoundException	El archivo de origen o la carpeta de destino no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.
IllegalParameterException	El recurso especificado a suprimir es una carpeta.

Ejemplo

El ejemplo siguiente suprime la versión del archivo *MyReport.rptdesign* etiquetada como *Test* del repositorio.

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

El método deleteFolder

Suprime una carpeta y su contenido del repositorio.

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

Tabla 27. Parámetros de entrada de deleteFolder.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto de la carpeta en el repositorio	"/Temp Folder" o "0a58c3670016a7860000010dcee0eaa28219"
<i>submittedHierarchy</i>	Opcional	Booleano	Indica si la carpeta se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 28. Valor de retorno de deleteFolder.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 29. Excepciones de deleteFolder.

Tipo	Descripción
ResourceNotFoundException	La carpeta especificada no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.
IllegalParameterException	El recurso especificado a suprimir no es una carpeta.

Ejemplo

El ejemplo siguiente suprime la carpeta denominada *Drafts* (borradores) del repositorio. Si se produce un problema al suprimir la carpeta, se envía un mensaje de excepción a la consola.

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

El método downloadFile

Descarga la versión específica de un archivo del repositorio al sistema de archivos local.

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

Tabla 30. Parámetros de entrada de downloadFile.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa de repositorio del archivo que se debe descargar	"/Temp Folder/Temp.txt"
<i>target</i>	Obligatorio	Cadena	Ruta completa (en el sistema de archivos local) de la carpeta en la que se va a descargar el archivo	"C:\Temp"

Tabla 30. Parámetros de entrada de `downloadFile` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>version</i>	Opcional. Se puede especificar la versión (<i>version</i>) o la etiqueta (<i>label</i>).	Cadena	Versión específica del archivo que se desea descargar	"0:2006-08-25 21:15:49.453"
<i>label</i>	Opcional. Se puede especificar la versión (<i>version</i>) o la etiqueta (<i>label</i>).	Cadena	Versión etiquetada específica del archivo que se desea descargar	"Version 2"

Tabla 31. Valor de retorno de `downloadFile`.

Tipo	Descripción
Resource	Contenedor de información sobre un objeto de repositorio. Consulte el tema "La clase Resource" en la página 53 para obtener más información. .

Tabla 32. Excepciones de `downloadFile`.

Tipo	Descripción
ResourceNotFoundException	El archivo de origen o la carpeta de destino no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente descarga una versión etiquetada como *Production* del archivo *MyReport.rptdesign* en el directorio *Shared* del sistema de archivos 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")
```

El método `exportResource`

Exporta una carpeta de repositorio especificada en un archivo de exportación *.pes designado en el sistema de archivos local.

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

Tabla 33. Parámetros de entrada de `exportResource`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa de repositorio o URI de objeto de la carpeta que se va a exportar	/Temp Folder o 0a58c3670016a78 60000010dcee0eaa28219
<i>target</i>	Obligatorio	Cadena	Ruta completa (en el sistema de archivos local) y nombre de archivo a los que se va a exportar la carpeta	C:\Temp\backup.pes

Tabla 34. Valor de retorno de `exportResource`.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 35. Excepciones de `exportResource`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen o la carpeta de destino no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.
<code>IllegalParameterException</code>	El destino especificado es una carpeta. El destino debe ser un archivo *.pes.

Ejemplo

El ejemplo siguiente exporta el contenido de la carpeta *Drafts* (borradores) en la carpeta *backups* (copias de seguridad) del sistema de archivos local.

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

El método `getAccessControlList`

Recupera la lista de control de acceso (ACL) de seguridad para el archivo o carpeta especificado del repositorio.

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

Tabla 36. Parámetros de entrada de `getAccessControlList`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	Opcional	Booleano	Indica si el archivo o carpeta se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 37. Valor de retorno de `getAccessControlList`.

Tipo	Descripción
Dictionary	Se muestra un diccionario que contiene el nombre de usuario y el permiso asociado. Por ejemplo: { "admin": "MODIFY_ACL", "Joe": "DELETE" }

Tabla 38. Excepciones de `getAccessControlList`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen o la carpeta de destino no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente imprime la ACL para el archivo *MyReport.rptdesign*.

```

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

```

El método getAllVersions

Recupera una lista de todas las versiones de un archivo en el repositorio.

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

Tabla 39. Parámetros de entrada de getAllVersions.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000010dcee0eaa28219"
<i>submittedHierarchy</i>	Opcional	Booleano	Indica si el archivo se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 40. Valor de retorno de getAllVersions.

Tipo	Descripción
List	Un lista de objetos de recursos. Consulte "La clase Resource" en la página 53.

Tabla 41. Excepciones de getAllVersions.

Tipo	Descripción
ResourceNotFoundException	El archivo de origen no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.
IllegalParameterException	El origen especificado es una carpeta.

Ejemplo

Este ejemplo recupera información sobre todas las versiones del archivo *MyReport.rptdesign*, imprimiendo el autor, marcador de versión y etiquetas de versión de cada una de ellas.

```

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()

```

El método getChildren

Recupera una lista de todos los archivos y carpetas en una carpeta de repositorio especificada.

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

Tabla 42. Parámetros de entrada de getChildren.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto de la carpeta en el repositorio	"/Temp Folder" o "0a58c3670016a7860000010dcee0eaa28219"

Tabla 42. Parámetros de entrada de `getChildren` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>submittedHierarchy</code>	Opcional	<i>Boolean</i>	Indica si la carpeta se encuentra en la carpeta Trabajos enviados	True o False

Tabla 43. Valor de retorno de `getChildren`.

Tipo	Descripción
List	Un lista de objetos de recursos. Consulte “La clase Resource” en la página 53.

Tabla 44. Excepciones de `getChildren`.

Tipo	Descripción
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.
<code>ResourceNotFoundException</code>	La carpeta no existe.

Ejemplo

El ejemplo siguiente recupera el contenido de la carpeta `/Demo/Drafts`, imprimiendo el título, autor e identificador de recursos de cada elemento.

```
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()
```

El método `getCustomPropertyValue`

Recupera los valores válidos aceptados por una propiedad personalizada especificada.

```
getCustomPropertyValue(propertyName)
```

Tabla 45. Parámetros de entrada de `getCustomPropertyValue`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>propertyName</code>	Obligatorio	Cadena	El nombre de la propiedad personalizada.	"FreeForm"

Tabla 46. Valor de retorno de `getCustomPropertyValue`.

Tipo	Descripción
List	Devuelve una lista de valores válidos que acepta la propiedad personalizada. Si la propiedad requiere una selección (por ejemplo, selección única o múltiple), la lista contiene todos los valores válidos para la selección. Si se trata de una propiedad de formulario libre, la lista contiene el tipo de datos que acepta la propiedad (por ejemplo, String, Date o Number).

Tabla 47. Excepciones de `getCustomPropertyValue`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	La propiedad especificada no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente accede a los valores de la propiedad personalizada `Language` (idioma).

```

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

```

El método getMetadata

Recupera los atributos de metadatos de un archivo o carpeta en el repositorio, incluyendo cualquier propiedad personalizada e información de tema.

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

Tabla 48. Parámetros de entrada de getMetadata.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000010dcee0eaa28219"
<i>version</i>	Opcional. Se puede especificar la versión (version) o la etiqueta (label).	Cadena	Versión específica del archivo o carpeta	"0:2006-08-25 21:15:49.453"
<i>label</i>	Opcional. Se puede especificar la versión (version) o la etiqueta (label).	Cadena	Versión etiquetada específica del archivo o carpeta	"Version 1"
<i>submittedHierarchy</i>	Opcional	Booleano	Indica si el archivo se encuentra en la carpeta <i>Submitted Jobs</i> (trabajos enviados)	True o False

Tabla 49. Valor de retorno de getMetadata.

Tipo	Descripción
Resource	Contenedor de información sobre un objeto de repositorio. Consulte el tema "La clase Resource" en la página 53 para obtener más información.

Tabla 50. Excepciones de getMetadata.

Tipo	Descripción
ResourceNotFoundException	El archivo o carpeta de origen no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente accede al identificador de recursos de la carpeta */Demo/Drafts*.

```

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

```

El método importResource

Importa un archivo de exportación *.pes existente del sistema de archivo local en el repositorio.

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

Tabla 51. Parámetros de entrada de importResource.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa (en el sistema de archivos local) del archivo que se va a importar	"C:\Temp\New.pes"
<i>target</i>	Obligatorio	Cadena	Ruta completa de repositorio o URI de objeto de la carpeta en la que se va a realizar la importación	"/Temp Folder" o "0a58c3670016a7860000010dcee0eaa28219"
<i>resourceType</i>	Opcional	Cadena	Tipo del contenido que se va a importar. Especifique uno de los valores siguientes: <ul style="list-style-type: none"> • ContentRepository para objetos de contenido como archivos y carpetas • ResourceDef para definiciones de recursos • Credential para credenciales de usuario • DataSource para definiciones de orígenes de datos • MessageDomain para dominios de mensajes • ServerCluster para definiciones de clúster de servidores • Server para definiciones de servidor • PromotionPolicy para políticas de promoción Si no se especifica este parámetro, se utilizará el valor predeterminado de ContentRepository .	

Tabla 51. Parámetros de entrada de `importResource` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>resourceConflict</code>	Opcional	Cadena	<p>Indica cómo resolver conflictos de ID o nombre duplicados. Especifique uno de los valores siguientes:</p> <ul style="list-style-type: none"> • keepTarget. Se mantendrá el elemento de destino. El elemento de origen con el ID duplicado, contenido en el archivo <code>.pes</code>, no se tendrá en cuenta. • addNewVersion. Esta opción se suele utilizar para resolver conflictos de ID o conflictos de nombre. Si se produce un conflicto de ID duplicado entre el objeto de origen y de destino, se creará una nueva versión del objeto en la ubicación de destino. Si se produce un conflicto de nombre, se cambiará el nombre del objeto importado en la ubicación de destino. Normalmente, a los objetos cuyos nombres se cambian se les añade <code>_1</code>, <code>_2</code> y así sucesivamente. En el caso de que un evento con dos versiones de objeto tengan la misma etiqueta, el sistema mantiene una etiqueta y desecha la etiqueta duplicada porque dos versiones del mismo elemento no pueden tener la misma etiqueta. La etiqueta que se conserva depende del parámetro labelFrom. <p>Si no se especifica este parámetro, se utilizará el valor predeterminado de keepTarget.</p>	
<code>labelFrom</code>	Opcional	Cadena	<p>Etiqueta a utilizar si dos versiones de un objeto tienen la misma etiqueta. Se descarta la etiqueta de la otra versión. Especifique source o target. Si no se especifica este parámetro, se utilizará el valor predeterminado de source.</p>	

Tabla 51. Parámetros de entrada de `importResource` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>lockResolution</code>	Opcional	Cadena	<p>Define cómo proceder si se detectan recursos bloqueados. Especifique uno de los valores siguientes:</p> <ul style="list-style-type: none"> • continue. La importación continúa, omitiendo los recursos bloqueados. • abort. Termina el proceso de importación si se encuentran recursos bloqueados. Si se encuentran conflictos debidos a bloqueos de objetos, el proceso de importación terminará y fallará. <p>Si no se especifica este parámetro, se utilizará el valor predeterminado de abort.</p>	
<code>invalidVersionConflict</code>	Opcional	Cadena	<p>Define cómo proceder si se detecta una versión no válida durante el proceso de importación. Especifique uno de los valores siguientes:</p> <ul style="list-style-type: none"> • import. Se importará la versión válida. • discard. La versión no válida se eliminará. <p>Si no se especifica este parámetro, se utilizará el valor predeterminado de import.</p>	

Tabla 51. Parámetros de entrada de `importResource` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>resourceDef</code>	Opcional	Cadena	<p>Define el comportamiento de proceso para definiciones de recursos. Especifique uno de los valores siguientes:</p> <ul style="list-style-type: none"> • recommended. Una definición de recurso sólo se importa si el identificador o nombre no entra en conflicto con una definición de destino. Las definiciones de recursos que tengan algún conflicto no se importan. • include. Se importan todas las definiciones de recursos del archivo de importación. Puede seleccionar uno o más tipos de definiciones de recursos para excluirlos de la importación marcando el recuadro de selección correspondiente. • exclude. No se importan definiciones de recursos desde los archivos de importación. Los objetos importados podrían necesitar modificación para hacer referencia a definiciones de recursos disponibles. <p>Si no se especifica este parámetro, se utilizará el valor predeterminado de recommended.</p>	

Tabla 51. Parámetros de entrada de `importResource` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>exclude</code>	Opcional	Cadena	<p>Define qué tipos de recursos se excluyen durante la importación. Se pueden combinar varios valores en cualquier orden como una lista delimitada por el carácter de punto y coma. Especifique uno o más de los valores siguientes:</p> <ul style="list-style-type: none"> • credential excluye credenciales de usuario • customproperty excluye propiedades personalizadas para objetos de recursos • datasource excluye definiciones de orígenes de datos • messagedomain excluye dominios de mensajes • notification excluye definiciones de notificaciones • servercluster excluye definiciones de clúster de servidores • server excluye definiciones de servidor • topic excluye definiciones de tema <p>Si no se especifica este parámetro, se incluyen todos los tipos en la importación.</p>	

Tabla 52. Valor de retorno de `importResource`.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 53. Excepciones de `importResource`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen o la carpeta de destino no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente importa el contenido del archivo de exportación `drafts pes` en la carpeta `/Demo/Drafts` del Repositorio de IBM SPSS Collaboration and Deployment Services

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

El método `moveResource`

Mueve un archivo o carpeta a otra carpeta del repositorio. Es posible cambiar el nombre de un archivo de origen especificado al moverlo, siendo la existencia y el tipo de destino lo que determina el nombre final.

En la tabla siguiente se describe el comportamiento de la función de cambio de nombre al mover un archivo:

Tabla 54. Cambio de nombre de archivos.

Tipo de destino	La carpeta de destino existe	La carpeta de destino no existe
carpeta	El archivo de origen pasa a ser un hijo de la carpeta de destino.	El archivo de origen se mueve a la carpeta padre del destino especificado y se cambia su nombre al nombre de la carpeta de destino.
archivo	El archivo de origen se mueve a la carpeta que contiene el archivo de destino y se cambia su nombre al nombre del destino.	Error notificado.

Por ejemplo, si el origen es el archivo `/Temp Folder/Temp.txt` y el destino especificado es la carpeta `/Demo Folder`, se podrían producir los resultados siguientes:

- Si la carpeta `Demo Folder` existe, `Temp.txt` se traslada a `Demo Folder`.
- Si la carpeta `Demo Folder` no existe, `Temp.txt` se mueve a `/"` y se cambia su nombre a `Demo Folder`.

O bien, si el origen es `/Temp Folder/Temp.txt` y el destino especificado es el archivo `/Demo Folder/Abc.dat`, se podrían producir los resultados siguientes:

- Si la carpeta `Demo Folder` existe, `Temp.txt` se traslada a `Demo Folder` y su nombre se cambia a `Abc.dat`.
- Si la carpeta `Demo Folder` no existe, se muestra un error.

`moveResource(source, target)`

Tabla 55. Parámetros de entrada de `moveResource`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	<code>/Temp Folder/Temp.txt</code> o <code>0a58c3670016a786000010dcee0eaa28219</code>
<code>target</code>	Obligatorio	Cadena	Ruta completa o URI de objeto de la carpeta a la que mover el archivo. También es posible indicar un nuevo nombre de archivo para cambiar el nombre del archivo o la carpeta especificada cuando se mueve.	<code>/New Folder</code> o <code>/New Folder/abc.dat</code>

Tabla 56. Valor de retorno de `moveResource`.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 57. Excepciones de `moveResource`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El origen especificado no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente mueve el archivo `MyReport.rptdesign` de la carpeta `/Demo/Drafts` a la carpeta `/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
```

El método `removeLabel`

Quita una etiqueta de un archivo del repositorio.

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

Tabla 58. Parámetros de entrada de `removeLabel`.

Campo	Uso	Tipo	Valor de ejemplo	Descripción
<code>source</code>	Obligatorio	Cadena	<code>"/Temp Folder/Temp.txt"</code> o <code>"0a58c3670016a7860000010dcee0eaa28219"</code>	Ruta completa o URI de objeto del archivo en el repositorio
<code>label</code>	Obligatorio	Cadena	<code>"Version 1"</code>	El nombre de etiqueta que se va a quitar

Tabla 59. Valor de retorno de `removeLabel`.

Tipo	Descripción
String	URI del archivo actualizado

Tabla 60. Excepciones de `removeLabel`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente elimina la etiqueta `Draft` del archivo `MyReport.rptdesign`.

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

El método `removeSecurity`

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

Define la lista de control de acceso de seguridad (ACL) para un archivo o carpeta especificado del repositorio.

Tabla 61. Parámetros de entrada de `removeSecurity`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	<code>"/Temp Folder/Temp.txt"</code> o <code>"0a58c3670016a7860000010dcee0eaa28219"</code>

Tabla 61. Parámetros de entrada de `removeSecurity` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>principal</i>	Obligatorio	Cadena	El usuario (por ejemplo, <i>admin</i>) a eliminar del archivo o carpeta especificado	administrador
<i>provider</i>	Opcional	Cadena	El proveedor de seguridad (como <i>Native</i>) que se utilizará para obtener la información sobre los usuarios. Son válidos los valores siguientes: <ul style="list-style-type: none"> • <i>Native</i> para el proveedor de seguridad local nativo intrínseco del sistema. Este es el proveedor predeterminado. • <i>AD_<nombre></i> para Active Directory, donde <i><nombre></i> corresponde al nombre de proveedor de seguridad dentro del sistema • <i>ADL_<nombre></i> para Active Directory con alteración temporal local, donde <i><nombre></i> corresponde al nombre del proveedor de seguridad dentro del sistema • <i>ldap_<nombre></i> para OpenLDAP, donde <i><nombre></i> corresponde al nombre del proveedor de seguridad dentro del sistema 	<i>Native</i>
<i>cascade</i>	Opcional	Booleano	Extiende los ajustes de seguridad a todos los archivos y subcarpetas de la carpeta especificada	True o False

Tabla 62. Valor de retorno de `removeSecurity`.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Tabla 63. Excepciones de `removeSecurity`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen o la carpeta de destino no existe.
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.
<code>IllegalParameterException</code>	El usuario o nombre de proveedor de seguridad especificado es incorrecto.

Ejemplo

El ejemplo siguiente elimina la ACL de un principal del archivo *MyReport.rptdesign*.

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

El método search

Busca archivos en el repositorio, devolviendo una lista de versiones de archivo con contenido de metadatos que coincide con los criterios de búsqueda.

```
search(criteria)
```

Tabla 64. Parámetros de entrada de search.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
criteria	Obligatorio	Cadena	El valor utilizado para buscar metadatos de archivos	"Age"

Tabla 65. Valor de retorno de search.

Tipo	Descripción
PageResult	Estructura en la que cada fila corresponde a una coincidencia de búsqueda. Consulte el tema "La clase PageResult" en la página 55 para obtener más información.

Tabla 66. Excepciones de search.

Tipo	Descripción
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente busca versiones de archivo que tienen el texto *Quarterly* en cualquier campo de los metadatos.

```
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()
```

El método setLabel

Aplica una etiqueta a una versión de un archivo en el repositorio. Si el archivo ya se ha etiquetado, la etiqueta original se reemplaza por una nueva.

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

Tabla 67. Parámetros de entrada de setLabel.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
source	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000 010dcee0eaa28219"
version	Obligatorio	Cadena	Versión específica del archivo	"0:2006-08-25 21:15:49.453"
label	Obligatorio	Cadena	La etiqueta que se aplicará al archivo	"Version 1"

Tabla 68. Valor de retorno de `setLabel`.

Tipo	Descripción
String	URI del archivo actualizado

Tabla 69. Excepciones de `setLabel`.

Tipo	Descripción
ResourceNotFoundException	El archivo o versión de origen no existe.
InsufficientParameterException	No se han especificado parámetros obligatorios.

Ejemplo

El ejemplo siguiente asigna la etiqueta *Beta* a la segunda versión del archivo *MyReport.rptdesign*. El método `getVersionMarker` para un objeto `Resource` devuelve el marcador de la versión que se va a etiquetar.

```
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")
```

El método `setMetadata`

Aplica las propiedades de metadatos a los archivos y carpetas del repositorio.

La tabla siguiente identifica las propiedades de metadatos y si se pueden aplicar a archivos y carpetas.

Tabla 70. Propiedades del objeto de repositorio.

Propiedad de metadatos	Tipo de recurso
Autor	Archivo
Descripción	Archivo o carpeta
Título	Archivo o carpeta
Fecha caducidad	Archivo o carpeta
Palabra clave	Archivo
Temas	Archivo
Metadatos personalizados	Archivo o carpeta

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

Tabla 71. Parámetros de entrada de `setMetadata`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa o URI de objeto del archivo o carpeta en el repositorio	"/Temp Folder/Temp.txt" o "0a58c3670016a7860000 010dcee0eaa28219"
<i>version</i>	Opcional. Se puede especificar la versión (version) o la etiqueta (label).	Cadena	Versión específica del archivo que se desea descargar	"0:2006-08-25 21:15:49.453"

Tabla 71. Parámetros de entrada de `setMetadata` (continuación).

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>label</i>	Opcional. Se puede especificar la versión (version) o la etiqueta (label).	Cadena	Etiqueta de la versión específica	"Label 1"
<i>props</i>	Obligatorio	Diccionario	Contiene todos los metadatos que se definen en el diccionario con el nombre de metadatos como claves. Como se muestra en la columna Valor de ejemplo, toma la lista como un valor de topic y diccionario para customProperty. Para el resto de los metadatos toma una cadena.	<pre>{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }</pre>

Tabla 72. Valor de retorno de `setMetadata`.

Tipo	Descripción
String	URI del archivo o carpeta para el que se definieron los metadatos

Tabla 73. Excepciones de `setMetadata`.

Tipo	Descripción
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.
<code>ResourceNotFoundException</code>	El archivo o carpeta de origen no existe.

Ejemplo

El ejemplo siguiente asigna la palabra clave *Quarterly* a la versión *Production* del archivo *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
```

El método `uploadFile`

Guarda un archivo en el repositorio del sistema de archivos local, con la opción de crear una nueva versión del archivo si éste ya existe.

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


Tabla 74. Parámetros de entrada de `uploadFile`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa (en el sistema de archivos local) del archivo que se desea cargar.	"C:\Temp\Temp.txt"
<code>target</code>	Obligatorio	Cadena	La ruta completa de la carpeta de destino en el repositorio donde se va a cargar el archivo	"/Temp Folder"
<code>versionFlag</code>	Opcional	Booleano	Si el archivo especificado ya existe, se creará una nueva versión del archivo.	True o False

Tabla 75. Valor de retorno de `uploadFile`.

Tipo	Descripción
String	URI del archivo cargado

Tabla 76. Excepciones de `uploadFile`.

Tipo	Descripción
<code>ResourceNotFoundException</code>	El archivo de origen o la carpeta de destino no existe.
<code>ResourceAlreadyExistsException</code>	Ya existe un archivo o carpeta con el mismo nombre que el archivo de origen en la carpeta de destino y no se ha especificado el parámetro <code>versionFlag</code> .
<code>InsufficientParameterException</code>	No se han especificado parámetros obligatorios.

Ejemplo

Este ejemplo carga el archivo `MyReport.rptdesign` en la carpeta `/Demo/Drafts` del repositorio. Si el archivo ya existe, se carga una nueva versión del archivo utilizando el parámetro `versionFlag`.

```
from pes.util.PESExceptions 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
```

Clases de derivador

La API `PESImpl` incluye clases que sirven como derivadores para los objetos devueltos desde los servicios web llamados por los métodos del repositorio de contenido. Estas clases de derivador proporcionan una interfaz para mostrar la información devuelta por los métodos.

La clase `Resource`

La clase `Resource` actúa como un derivador simplificado del objeto de repositorio `ResourceSpecifier.Resource`, ofreciendo acceso a información específica del objeto.

Además de los metadatos estándares asociados con los objetos de repositorio, esta clase incluye la información de metadatos personalizados definida para los objetos del repositorio. Tabla 77 en la página 54 enumera todos los métodos disponibles en la clase `Resource`.

Tabla 77. Métodos de la clase Resource.

Nombre del método	Descripción
getAccessControlList	Devuelve un diccionario de los permisos de seguridad de un objeto. Contiene el nombre de usuario como clave y sólo el permiso más alto ofrecido al usuario. Por ejemplo: Si el usuario <i>Joe</i> tiene el permiso <i>delete</i> (suprimir) sobre el recurso <i>X</i> , <code>getAccessControlList</code> sobre el objeto de recurso que representa <i>X</i> devolverá <code>{'Joe': 'DELETE'}</code> y no los tres permisos (<i>read</i> , <i>write</i> , <i>delete</i>) de la llamada de servicio web.
getOwner	Devuelve el nombre del propietario del objeto como una cadena
getAuthor	Devuelve el nombre del autor del objeto como una cadena
getContentSize	Devuelve el tamaño del objeto
getCreatedBy	Devuelve el nombre del usuario que creó el objeto como una cadena
getCreationDate	Devuelve la fecha de creación del objeto como objeto de fecha y hora
getDescription	Devuelve la descripción del objeto como una lista
getDescriptionLanguage	Devuelve el idioma del objeto como una lista
getExpirationDate	Devuelve la fecha de caducidad del objeto como objeto de fecha y hora
isExpired	Indica si el objeto especificado ha caducado o no
getMIMEType	Devuelve el tipo MIME del objeto como una cadena
getModificationDate	Devuelve la fecha de última modificación del objeto como objeto de fecha y hora
getObjectCreationDate	Devuelve la fecha de creación del objeto como objeto de fecha y hora
getObjectLastModifiedBy	Devuelve el usuario que modificó por última vez el objeto como una cadena
getObjectLastModifiedDate	Devuelve la fecha de última modificación del objeto como objeto de fecha y hora
getResourceID	Devuelve el identificador de recurso del objeto como una cadena
getResourcePath	Devuelve la ruta del objeto especificado como una cadena
getTitle	Devuelve el título del objeto como una cadena
getTopicList	Devuelve la lista de temas del objeto
getVersionMarker	Devuelve la versión del objeto como una cadena
getVersionLabel	Devuelve la etiqueta del objeto como una cadena
getCustomMetadata	Devuelve cualquier propiedad personalizada asociada con el objeto como un diccionario
getKeywordList	Devuelve una lista de palabras clave asociadas con el objeto

La clase IdentificationSpecifier

Esta clase actúa como un derivador simplificado al objeto de repositorio `IdentificationSpecifier`, permitiendo el acceso a datos específicos de identificación para el objeto.

Tabla 78 enumera todos los métodos disponibles en la clase `IdentificationSpecifier`.

Tabla 78. Métodos de la clase IdentificationSpecifier.

Nombre del método	Descripción
getIdentifier	Devuelve el valor de identificador del objeto como una cadena
getVersionMarker	Devuelve la versión del objeto como una cadena

Tabla 78. Métodos de la clase *IdentificationSpecifier* (continuación).

Nombre del método	Descripción
getVersionLabel	Devuelve la etiqueta aplicada a una versión de objeto como una cadena

La clase **PageResult**

Esta clase `PageResult` sirve como contenedor de resultados de búsqueda. Una coincidencia individual corresponde a una fila del objeto `PageResult`.

Por ejemplo, una búsqueda que devuelve cuatro recursos corresponderá a un objeto `PageResult` que contiene cuatro filas. Tabla 79 enumera todos los métodos disponibles en la clase `PageResult`.

Tabla 79. Métodos de la clase *PageResult*.

Nombre del método	Descripción
getRows	Devuelve una lista de objetos <code>SearchRow</code> . Consulte el tema “La clase <code>SearchRow</code> ” para obtener más información.

La clase **SearchRow**

La clase `SearchRow` sirve como un contenedor de información a nivel de objetos sobre un resultado de búsqueda individual. Puede acceder a los metadatos sobre un objeto utilizando los métodos de esta clase.

Tabla 80 muestra todos los métodos disponibles de la clase `SearchRow`.

Tabla 80. Métodos de la clase *SearchRow*.

Nombre del método	Descripción
getTitle	Devuelve el nombre del archivo o carpeta
getAuthor	Devuelve el autor del archivo o carpeta
getMIMEType	Devuelve el tipo MIME del archivo
getObjectLastModifiedBy	Devuelve el usuario que modificó el archivo o carpeta por última vez
getModified	Devuelve la fecha y hora a las que se modificó por última vez el archivo o carpeta
getFolderPath	Devuelve la ubicación del archivo o carpeta
getFolder	Devuelve el nombre de la carpeta padre del archivo o carpeta
getParentURI	Devuelve el URI de objeto del padre
getTopic	Devuelve los temas asociados al archivo o carpeta
getChildRow	Devuelve la lista de objetos <code>SearchChildRow</code> (consulte la sección siguiente para obtener más información)

Para acceder a la información en el nivel de versión de un objeto, utilice el método `getChildRow` para devolver las filas hijas correspondientes a las versiones de objeto.

La clase **SearchChildRow**

La clase `SearchChildRow` sirve como un contenedor de información a nivel de versiones sobre un resultado de búsqueda individual. Puede acceder a los metadatos sobre la versión de un objeto utilizando los métodos de esta clase.

Tabla 81 en la página 56 muestra todos los métodos disponibles de la clase `SearchChildRow`.

Tabla 81. Métodos de la clase SearchChildRow.

Nombre del método	Descripción
getExpirationDate	Devuelve la fecha de caducidad del archivo o carpeta
getKeyword	Devuelve las palabras clave asociadas con la versión del archivo o carpeta
getVersionLabel	Devuelve la etiqueta de versión del archivo o carpeta
getDescription	Devuelve la descripción del archivo o carpeta
getLanguage	Devuelve el idioma
getVersionCreationDate	Devuelve la fecha y la hora en que se creó el archivo o carpeta
getVersionMarker	Devuelve el marcador de versión del archivo o carpeta
getUri	Devuelve el objeto URI del archivo o carpeta

API de gestión de procesos

Los scripts de gestión de procesos ofrecen la posibilidad de trabajar con trabajos. Esta área incluye las funciones siguientes:

- Ejecución de trabajos
- Recuperación de historiales de trabajos
- Recuperación de detalles de trabajos

En esta sección se describen los métodos PESImpl utilizados para trabajar con trabajos almacenados en el repositorio. Cada método contiene información detallada sobre sintaxis, un ejemplo y los mensajes esperados.

Métodos

Las secciones siguientes muestran todos los métodos de scripts de gestión de procesos compatibles con IBM SPSS Collaboration and Deployment Services.

Nota: para todos los métodos que requieren una ruta a los archivos/carpetas del repositorio, puede utilizarse la ruta o el URI del objeto. El URI de objeto puede obtenerse visualizando las propiedades de objeto en IBM SPSS Collaboration and Deployment Services Deployment Manager.

El método cancelJob

Cancela un trabajo en ejecución.

`cancelJob(executionId)`

Tabla 82. Parámetros de entrada de cancelJob.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>executionId</i>	Obligatorio	Cadena	ID de ejecución del trabajo	0a58c33d002ce90800 00010e0ccf7b01800e

Tabla 83. Valor de retorno de cancelJob.

Tipo	Descripción
Boolean	Devuelve un mensaje de éxito cuando se cancela el trabajo

Ejemplo

Este ejemplo termina la ejecución del trabajo *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

```

El método deleteJobExecutions

Suprime una o más ejecuciones de trabajos.

`deleteJobExecutions(executionId)`

Tabla 84. Parámetros de entrada de deleteJobExecutions.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>executionId</i>	Obligatorio	Lista	Lista de identificadores de ejecución de trabajos	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce9080060010e0cdf7b01802r]

Tabla 85. Valor de retorno de deleteJobExecutions.

Tipo	Descripción
Boolean	True o False, en función de si el método se ejecuta correctamente.

Ejemplo

Este ejemplo suprime las ejecuciones del trabajo *Reports* (Informes).

```

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

# Obtener el ID de ejecución del historial de ejecución
deleteList = []
for exrow in execRows :
    uuid = exrow.getEventObjId()
    deleteList.append(uuid)

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

```

El método executeJob

Ejecuta un trabajo de forma síncrona o asíncrona en función de los parámetros pasados. En el caso de una ejecución asíncrona, el método no devuelve el control hasta que se completa el trabajo. En el caso de una ejecución síncrona, el método devuelve el control después de que se inicie el trabajo.

`executeJob(source,notification,asynchronous)`

Tabla 86. Parámetros de entrada de executeJob.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>source</i>	Obligatorio	Cadena	Ruta completa (en el sistema de archivos local) del archivo que se desea cargar.	"C:\Temp\Temp.txt"
<i>notification</i>	Opcional	Booleano	Indica si el trabajo se ejecuta con notificaciones o sin ellas. El valor predeterminado es False.	True o False
<i>asynchronous</i>	Opcional	Booleano	Indica si el trabajo se ejecuta de forma asíncrona. El valor predeterminado es False.	True o False

Tabla 87. Valor de retorno de `executeJob`.

Tipo	Descripción
String	El identificador exclusivo de la ejecución del trabajo. Este identificador se utiliza para hacer referencia a la ejecución del trabajo concreta.

Ejemplo

Este ejemplo inicia la ejecución del trabajo *Reports* de forma asíncrona con notificaciones.

```
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
```

El método `getJobExecutionDetails`

Enumera los detalles de ejecución de un trabajo específico, incluyendo cualquier paso e iteración del trabajo.

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

Tabla 88. Parámetros de entrada de `getJobExecutionDetails`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<i>executionId</i>	Obligatorio	Cadena	ID de ejecución del trabajo	0a58c33d002ce9080000 010e0ccf7b01800e
<i>log</i>	Opcional	Booleano	Indica si el registro del trabajo se muestra en línea	True o False
<i>target</i>	Opcional	Cadena	La ubicación (en el sistema de archivos local) para almacenar los registros. Sólo se usa en conjunto con el parámetro <code>--log</code> .	"c:\logs"

Tabla 89. Valor de retorno de `getJobExecutionDetails`.

Tipo	Descripción
<code>jobExecutionDetails</code>	Detalles sobre una ejecución de trabajos. Consulte el tema "La clase <code>jobExecutionDetails</code> " en la página 60 para obtener más información.

Ejemplo

Este ejemplo recupera información sobre las ejecuciones de pasos de trabajo para la ejecución de trabajos que tiene el identificador *execId*, enviando el resultado de cada paso a la consola.

```
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()
```

El método `getJobExecutionList`

Enumera las ejecuciones de un trabajo específico, incluyendo cualquier trabajo en ejecución y los trabajos completados para todas las versiones del trabajo.

```
getJobExecutionList(source)
```

Tabla 90. Parámetros de entrada de `getJobExecutionList`.

Campo	Uso	Tipo	Descripción	Valor de ejemplo
<code>source</code>	Obligatorio	Cadena	Ruta completa del trabajo en el repositorio.	<code>"/testJob"</code>

Tabla 91. Valor de retorno de `getJobExecutionList`.

Tipo	Descripción
<code>PageResult</code>	Contenedor de la lista de ejecuciones para un trabajo. Consulte el tema "La clase <code>PageResult</code> " para obtener más información. .

Ejemplo

Este ejemplo recupera las ejecuciones del trabajo `Reports`, enviando información sobre cada ejecución a la consola.

```
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()
```

Clases de derivador

La API `PESImpl` incluye clases que sirven como derivadores para los objetos devueltos desde los servicios web llamados por los métodos de gestión de procesos. Estas clases de derivador proporcionan una interfaz para mostrar la información devuelta por los métodos.

La clase `PageResult`

La clase `PageResult` sirve como un contenedor de resultados de ejecución de trabajos, permitiendo la recuperación de datos específicos de la ejecución de trabajos.

Una ejecución de trabajos individual corresponde a una fila del objeto `PageResult`. Por ejemplo, un trabajo que se haya ejecutado cuatro veces corresponderá a un objeto `PageResult` que contiene cuatro filas. Tabla 92 enumera todos los métodos disponibles en la clase `PageResult`.

Tabla 92. Métodos de la clase `PageResult`.

Nombre del método	Descripción
<code>getRows</code>	Devuelve una lista de objetos <code>Row</code> , representando cada uno de ellos una ejecución de un trabajo. Consulte el tema "La clase <code>Row</code> " para obtener más información.

La clase `Row`

La clase `Row` sirve como un contenedor de información a nivel de trabajos sobre una ejecución de trabajos. Puede acceder a los metadatos sobre una ejecución de trabajos utilizando los métodos de esta clase.

Tabla 93 enumera todos los métodos disponibles en la clase `Row`.

Tabla 93. Métodos de la clase `Row`.

Nombre del método	Descripción
<code>getObjId</code>	Devuelve el ID de ejecución del trabajo
<code>getPath</code>	Devuelve la ruta del trabajo

Tabla 93. Métodos de la clase Row (continuación).

Nombre del método	Descripción
getVersionMarker	Devuelve el marcador de versión del trabajo que se estaba ejecutando
getVersionLabel	Devuelve la etiqueta de versión del trabajo que se estaba ejecutando
getEventObjId	Devuelve el ID de evento del trabajo que se estaba ejecutando
getEventState	Devuelve el estado del trabajo en ejecución
getEventCompletionCode	Devuelve el código de finalización del trabajo
getEventStartDateTime	Devuelve la fecha y hora de inicio del trabajo
getEventEndDateTime	Devuelve la fecha y hora de finalización del trabajo
getQueuedDateTime	Devuelve la fecha y hora de puesta en cola del trabajo

La clase jobExecutionDetails

Es la clase que devuelve el método `getJobExecutionDetails`. Almacena los detalles de ejecución de un trabajo e incluye una lista de objetos `jobStepExecution` que proporcionan información sobre cada paso del trabajo.

Tabla 94 enumera todos los métodos disponibles en la clase `jobExecutionDetails`.

Tabla 94. Métodos de la clase `jobExecutionDetails`.

Nombre del método	Descripción
getJobStepDetails	Devuelve una lista de objetos <code>jobStepExecutionDetails</code> . Consulte el tema “La clase <code>jobStepExecutionDetails</code> ” en la página 61 para obtener más información.
getArtifactLocation	Devuelve una lista de las ubicaciones de artefactos de trabajo
getCompletionCode	Devuelve el código de finalización de la ejecución de trabajos
getEndDateTime	Devuelve la fecha y hora de finalización de la ejecución de trabajos
getEventName	Devuelve el nombre de evento de la ejecución de trabajos
getEventUUID	Devuelve el ID de evento de la ejecución de trabajos
getExecutionState	Devuelve el estado de ejecución de la ejecución de trabajos
getExecutionSuccess	Devuelve el estado de fallo o éxito de la ejecución de trabajos
getExecutionWarning	Indica si han aparecido advertencias
getLog	Devuelve el registro generado (como cadena)
getNotificationEnabled	Indica si las notificaciones por correo electrónico están o no activadas
getQueuedDateTime	Devuelve la fecha y hora de puesta en cola de la ejecución de trabajos
getStartDateTime	Devuelve la fecha y hora de inicio de la ejecución de trabajos
getUserName	Devuelve el nombre del usuario que ejecutó el trabajo
getUUID	Devuelve el ID de ejecución del trabajo

La clase `jobStepExecutionDetails`

Esta clase almacena los detalles de ejecución de un paso de trabajo y una lista de objetos `jobStepChildExecutionDetails`. Esta clase contiene el objeto `ExecutionDetails`, al que delega todas sus llamadas de método.

Tabla 95 enumera todos los métodos disponibles en la clase `jobStepExecutionDetails`.

Tabla 95. Métodos de la clase `jobStepExecutionDetails`.

Nombre del método	Descripción
<code>getJobStepChildExecutionList</code>	Devuelve una lista de objetos <code>jobStepChildExecutionDetails</code> . Consulte el tema “La clase <code>jobStepChildExecutionDetails</code> ” para obtener más información.
<code>getArtifactLocation</code>	Devuelve una lista de las ubicaciones de artefactos de paso de trabajo
<code>getCompletionCode</code>	Devuelve el código de finalización del paso de trabajo
<code>getEndTime</code>	Devuelve la fecha y hora de finalización del paso de trabajo
<code>getEventName</code>	Devuelve el nombre de evento del paso de trabajo
<code>getEventUUID</code>	Devuelve el ID de evento del paso de trabajo
<code>getExecutionState</code>	Devuelve el estado de ejecución del paso de trabajo
<code>getExecutionSuccess</code>	Devuelve el estado de fallo o éxito del paso de trabajo
<code>getExecutionWarning</code>	Indica si han aparecido advertencias
<code>getLog</code>	Devuelve el registro generado (como cadena)
<code>getNotificationEnabled</code>	Indica si las notificaciones por correo electrónico están o no activadas
<code>getQueuedDateTime</code>	Devuelve la fecha y hora de puesta en cola del paso de trabajo
<code>getStartTime</code>	Devuelve la fecha y hora de inicio del paso de trabajo
<code>getUserName</code>	Devuelve el nombre del usuario que ejecutó el paso de trabajo
<code>getUUID</code>	Devuelve el ID de ejecución del paso de trabajo

La clase `jobStepChildExecutionDetails`

La clase `jobStepChildExecutionDetails` sirve como un contenedor de ejecuciones hijo de pasos de trabajo individuales. Por ejemplo, un paso de trabajo de informe iterativo produce una ejecución hijo para cada iteración del paso. Puede acceder a los metadatos sobre las ejecuciones hijo utilizando los métodos de esta clase.

Tabla 96 enumera todos los métodos disponibles en la clase `jobStepChildExecutionDetails`.

Tabla 96. Métodos de la clase `jobStepChildExecutionDetails`.

Nombre del método	Descripción
<code>getArtifactLocation</code>	Devuelve una lista de las ubicaciones de artefactos de ejecuciones hijo
<code>getCompletionCode</code>	Devuelve el código de finalización de la ejecución hijo
<code>getEndTime</code>	Devuelve la fecha y hora de finalización de la ejecución hijo
<code>getEventName</code>	Devuelve el nombre de evento de la ejecución hijo
<code>getEventUUID</code>	Devuelve el ID de evento de la ejecución hijo
<code>getExecutionState</code>	Devuelve el estado de ejecución de la ejecución hijo

Tabla 96. Métodos de la clase `jobStepChildExecutionDetails` (continuación).

Nombre del método	Descripción
<code>getExecutionSuccess</code>	Devuelve el estado de fallo o éxito de la ejecución hijo
<code>getExecutionWarning</code>	Indica si han aparecido advertencias
<code>getLog</code>	Devuelve el registro generado (como cadena)
<code>getNotificationEnabled</code>	Indica si las notificaciones por correo electrónico están activadas
<code>getQueuedDateTime</code>	Devuelve la fecha y hora de puesta en cola de la ejecución hijo
<code>getStartDateTime</code>	Devuelve la fecha y hora de inicio de la ejecución hijo
<code>getUserName</code>	Devuelve el nombre del usuario que ejecutó la ejecución hijo
<code>getUUID</code>	Devuelve el ID de ejecución de la ejecución hijo

Scripts de ejemplo

Hay scripts de ejemplo que ilustran el uso de la clase `PESImpl` instalados en el directorio siguiente:

`<ubicación de instalación>/samples`

Estos scripts realizan diversas tareas, incluyendo las siguientes:

- Supresión de elementos caducados del Repositorio de IBM SPSS Collaboration and Deployment Services
- Supresión de artefactos enviados caducados
- Supresión de historiales de trabajos

Puede invocar los scripts desde un paso de trabajo general en IBM SPSS Collaboration and Deployment Services Deployment Manager para realizar tareas de mantenimiento del repositorio.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU. Es posible que este material esté disponible en IBM en otros idiomas. Sin embargo, es posible que necesite ser el propietario de una copia del producto o de una versión del producto en ese idioma para poder acceder a él.

Es posible que IBM no ofrezca los productos, servicios o las características que se describen en este documento en otros países. Póngase en contacto con el representante local de IBM si desea obtener información sobre los productos y servicios disponibles en su zona. Las referencias hechas a un producto, programa o servicio IBM no pretenden afirmar ni dar a entender que sólo se puede utilizar dicho producto, programa o servicio IBM. En su lugar, puede utilizarse cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y comprobar el funcionamiento de todo producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas sobre las licencias, por escrito, a:

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

Para consultas de licencia relativas a la información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país, o envíe las consultas, por escrito, a:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍAS DE NINGUNA CLASE, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas jurisdicciones no permiten la renuncia de garantías expresas o implícitas en determinadas transacciones, por lo que esta declaración podría no ser aplicable a su caso.

Esta información podría contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información contenida en este documento; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM se reserva el derecho a realizar en cualquier momento y sin notificación previa, mejoras o modificaciones en los productos y programas que se describen en el presente manual.

Las referencias en esta información a sitios web que no son de IBM se proporcionan sólo por comodidad y de ningún modo suponen un aval de dichos sitios. Los materiales de esos sitios web no forman parte de los materiales de este producto de IBM y la utilización de esos sitios web es por cuenta y riesgo del usuario.

IBM puede utilizar o distribuir la información que se le proporcione del modo que considere adecuado sin incurrir por ello en ninguna obligación con el remitente.

Los usuarios con licencia de este programa que deseen tener información sobre éste con el fin de habilitar: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido este) y (ii) el uso mutuo de la información que se ha intercambiado, deben ponerse en contacto con:

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

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo en algunos casos el pago de una cantidad.

IBM proporciona el programa bajo licencia descrito en esta información y todo el material bajo licencia disponible para él de acuerdo con los términos del Contrato del cliente de IBM, el Acuerdo internacional de licencia de programas de IBM o cualquier contrato equivalente existente entre las partes.

Los ejemplos de cliente y de datos de rendimiento citados se presentan solamente a efectos ilustrativos. Los resultados reales de rendimiento pueden variar en función de las configuraciones específicas y las condiciones de operación.

La información relacionada con productos no IBM se ha obtenido de los proveedores de dichos productos, de sus anuncios publicados o de otras fuentes disponibles de disponibilidad pública. IBM no ha comprobado esos productos y no puede confirmar la precisión del rendimiento, la compatibilidad o cualquier otra afirmación relacionada con productos que no son de IBM. Las preguntas relacionadas con las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Las declaraciones relativas a la orientación o intención futura de IBM están sujetas a cambio o se pueden retirar sin previo aviso y representan solamente metas y objetivos.

Este manual contiene ejemplos de datos e informes que se utilizan en operaciones empresariales cotidianas. Para ilustrarlos de la forma más completa posible, se han utilizado nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres de personas o empresas reales es mera coincidencia.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir los programas de muestra de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de muestra. Estos ejemplos no se han probado exhaustivamente en todas las condiciones. IBM, por tanto, no puede garantizar ni dar por supuesta la fiabilidad, la capacidad de servicio ni la funcionalidad de estos programas. Los programas de muestra se proporcionan "TAL CUAL" sin garantía de ningún tipo. IBM no será responsable de ningún daño resultante de la utilización de los programas de muestra por parte del usuario.

Consideraciones sobre la política de privacidad

Los productos de software de IBM, incluido el software tal como soluciones de servicio ("Ofertas de software"), pueden utilizar cookies u otras tecnologías para recopilar información de uso del producto a fin de ayudar a mejorar la experiencia del usuario final y personalizar las interacciones con el usuario final o para otros fines. En muchos casos, las ofertas de software no recopilan información de identificación personal. Alguna de nuestras ofertas de software puede ayudarle a ser capaz de recopilar

información personal identificable. Si esta oferta de Software usa cookies para recopilar información personal identificable, la información específica sobre el uso de cookies por esta oferta se expone más abajo.

Esta Oferta de software no utiliza cookies ni otras tecnologías para recopilar información de identificación personal.

Si las configuraciones desplegadas para esta oferta de software le ofrecen como cliente la posibilidad de recopilar información de identificación personal de los usuarios finales mediante cookies y otras tecnologías, debe buscar asesoramiento legal sobre las leyes aplicables a dicha recopilación de datos, incluidos los requisitos de aviso y consentimiento.

Para obtener más información sobre el uso de las diversas tecnologías, incluidas las cookies, para estos fines, consulte la política de privacidad de IBM en <http://www.ibm.com/privacy> y la declaración de privacidad en línea de IBM en <http://www.ibm.com/privacy/details>, la sección "Cookies, Web Beacons and Other Technologies" e "IBM Software Products and Software-as-a-Service Privacy Statement" en <http://www.ibm.com/software/info/product-privacy>.

Marcas comerciales

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de producto y servicio pueden ser marcas registradas de IBM u otras compañías. Hay disponible una lista actualizada de las marcas registradas de IBM en la web, en "Copyright and trademark information", en www.ibm.com/legal/copytrade.shtml.

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript son marcas registradas de Adobe Systems Incorporated en los Estados Unidos o en otros países.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas comerciales o marcas registradas de Intel Corporation o sus filiales en Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos, en otros países o bien ambas cosas.

UNIX es una marca registrada de The Open Group en EE.UU. y en otros países.

Java y todas las marcas comerciales y logotipos con base Java son marcas comerciales o son marcas registradas de Oracle y/o sus filiales.

Otros productos y nombres de servicio pueden ser marcas comerciales de IBM u otras empresas.



Impreso en España