

**IBM SPSS Collaboration and Deployment  
Services - Essentials for Python  
V 8 R 2**

**开发者指南**

**IBM**

**注**

在使用本资料及其支持的产品之前，请阅读第 57 页的『声明』中的信息。

**产品信息**

本版本适用于 IBM SPSS Collaboration and Deployment Services V8.2.1 及所有后续发行版，直到在新版本中另有声明为止。

**© Copyright IBM Corporation 2000, 2019.**

---

# 目录

## 第 1 章 IBM SPSS Collaboration and Deployment Services - Essentials for Python . . . . . 1

概述 . . . . .	1
安装 . . . . .	1
不推荐使用的功能. . . . .	1

## 第 2 章 命令行脚本编制 . . . . . 3

全局关键字 . . . . .	3
存储库连接 . . . . .	4
内容存储库脚本编制 . . . . .	4
关键字 . . . . .	4
内容存储库操作 . . . . .	5
流程管理功能. . . . .	21
关键字 . . . . .	21

流程管理操作. . . . .	21
-----------------	----

## 第 3 章 PESImpl 模块 . . . . . 25

内容存储库 API. . . . .	26
方法. . . . .	26
Wrapper 类 . . . . .	48
流程管理 API. . . . .	50
方法. . . . .	50
Wrapper 类 . . . . .	53
示例脚本 . . . . .	56

## 声明 . . . . . 57

隐私策略注意事项 . . . . .	58
商标. . . . .	58



---

# 第 1 章 IBM SPSS Collaboration and Deployment Services - Essentials for Python

---

## 概述

IBM® SPSS® Collaboration and Deployment Services 提供包含一组 API 的脚本编制框架，高级用户和管理员可用于编写独立例程或组合一组例程的批处理作业以供处理存储库对象和作业。这可大幅简化批量任务，包括以下任务：

- 更改一大组用户的安全许可权
- 为大量文件夹或文件添加标签或者除去其中的标签
- 上载或下载大量文件夹或文件

框架包含从命令行执行任务的能力，以及丰富的 API 以供与您自己的 Python 代码中的 IBM SPSS Collaboration and Deployment Services Repository 进行交互。

有关 Python（一种面向对象程序设计语言）的一般信息，请参阅 Python 站点。

---

## 安装

脚本编制框架可安装在 Windows 和 UNIX 平台上。脚本编制平台独立于脚本编制工具访问的存储库所使用的平台。

例如，在 UNIX 平台上运行的脚本编制函数可调用在 Windows 平台上运行的存储库。

有关安装指示信息，请参阅 IBM SPSS Collaboration and Deployment Services - Essentials for Python 安装指南。

---

## 不推荐使用的功能

如果要从 IBM SPSS Collaboration and Deployment Services 的先前发行版迁移，您应该了解自上一个版本以来已不推荐使用的各种功能。

如果不推荐某个功能，那么在产品的后续发行版中，IBM Corp. 可能除去此功能。未来的投资将以建议的迁移操作下面列出的策略功能作为重点。通常，除非提供了等效的备用功能，否则不会不推荐使用某项功能。

**此发行版中无不推荐使用的功能。** 针对参考目的，下表指示在产品的最新先前版本中不推荐使用的功能。此表还尽可能列出了建议的迁移操作。

表 1. 先前版本中不推荐的功能

不推荐使用	建议的迁移操作
安全提供程序：包含本地替换的 Active Directory，支持扩展组和允许的用户	使用已添加任何必需组的标准 Active Directory 安全提供程序
IBM SPSS Collaboration and Deployment Services Enterprise View	使用"分析数据视图"功能
IBM SPSS Collaboration and Deployment Services Enterprise View Driver	使用"分析数据视图"功能

表 1. 先前版本中不推荐的功能 (续)

不推荐使用	建议的迁移操作
方案文件	不再支持方案文件 (.scn)。无法在 Deployment Manager 中修改"企业视图"源节点。可在 IBM SPSS Modeler 客户机中修改旧方案文件，并将其重新保存为流文件。并且，必须删除并基于流文件重新创建使用方案文件的评分配置。
适用于 IBM SPSS Deployment Manager 的 Web 安装	使用独立安装程序
BIRT Report Designer for IBM SPSS	无
BIRT Report Designer for IBM SPSS 查看器	无
IBM SPSS Collaboration and Deployment Services Portlet	直接使用 IBM SPSS Collaboration and Deployment Services Deployment Portal，或者使用 Web service API
IBM SPSS Collaboration and Deployment Services Web Part	直接使用 IBM SPSS Collaboration and Deployment Services Deployment Portal，或者使用 Web service API
评分服务 V1 API	评分服务 V2 API
调度服务器服务	无
报告服务	无
认证服务 login 操作	认证服务 doLogin 操作
搜索服务 search 操作	搜索服务 search2.5 操作
SPSS AXIS/Castor Web service 客户机 jar	使用 Java 运行时环境、集成开发环境或 Eclipse Web Tools Platform (WTP) 随附的工具
clemrtl_setLogFile() API 函数	无

## 第 2 章 命令行脚本编制

可以从命令行使用 Python 文件 CADSTool.py 来处理 IBM SPSS Collaboration and Deployment Services Repository 中存储的资源。

用于从命令行调用 IBM SPSS Collaboration and Deployment Services 脚本编制操作的常规语法为：

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

其中：

- <Operation> 指定要调用的函数
- <Keywords> 定义用作函数的输入参数的键/值对

### 全局关键字

表 2 列出所有 IBM SPSS Collaboration and Deployment Services 脚本编制函数支持的关键字。第二列列出关键字的任何可选短版本。关键字区分大小写。

表 2. 全局关键字.

关键字	可选短版本	用途
--user	-u	用于连接到存储库服务器的用户名。如果用户不是来自于本机提供程序，那么值应包含表示安全提供程序的前缀。以下前缀值有效： <ul style="list-style-type: none"><li>• Native，针对于系统固有的本机本地安全提供程序。这是缺省提供程序。</li><li>• AD_&lt;name&gt;，针对于 Active Directory，其中 &lt;name&gt; 对应于系统中的安全提供程序名称</li><li>• ADL_&lt;name&gt;，针对于具有本地替换的 Active Directory，其中 &lt;name&gt; 对应于系统中的安全提供程序名称</li><li>• ldap_&lt;name&gt;，针对于 OpenLDAP，其中 &lt;name&gt; 对应于系统中的安全提供程序名称</li></ul> 前缀后跟斜杠和用户名。对于 Active Directory 提供程序，前缀中包含域。例如，对于 Active Directory 实例 AD_SPSSAD 的 MYDOMAIN 域中的用户 icrod，值为 AD_SPSSAD/MYDOMAIN/icrod。如果用户 icrod 位于 OpenLDAP 提供程序 SPSSLDAP 中，那么值为 ldap_SPSSLDAP/icrod。
--password	-p	用于连接到存储库服务器的密码
--host	-q	安装存储库的主机/服务器名称
--port	-o	存储库服务器端口号
--ssl		指示存储库服务器使用安全套接字层 (SSL) 协议来加密通信。如果使用此关键字，那么必须针对 SSL 配置存储库服务器。有关更多信息，请参阅管理员文档。
--server_url	-S	存储库服务器的完整 URL。如果服务器的 URL 包含定制上下文根，那么使用此关键字。如果指定服务器 URL，那么 host、port 和 ssl 关键字的值非必需。

表 2. 全局关键字 (续).

关键字	可选短版本	用途
<code>--useDefault</code>	<code>-z</code>	使用 <code>Authorization.properties</code> 文件中定义的服务器连接信息
<code>--help</code>	<code>-h</code>	脚本编制模块帮助信息

## 存储库连接

您必须在命令行尾部指定 IBM SPSS Collaboration and Deployment Services Repository 用户标识、密码和存储库服务器信息。

以下方法可用于提供此连接信息：

- 使用关键字，如以下示例中所示：

```
--user user --password password --host host --port port
--user user --password password --server_url url
```

- 通过 `Authorization.properties` 文件，其中命令行包含 `--useDefault` 参数（或短版本 `-z`）。此方法从位于 `Scripting folder\Lib\site-packages\config\Authorization.properties` 的 `Authorization.properties` 文件检索连接信息。使用简单文本编辑器来修改文件中的以下值以匹配您的存储库设置：

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

或者，您可以使用属性文件中的 `server_url` 关键字。

```
# Authorization Information
user=admin
password=pwd
server_url=http://yourhost:80/context_root
```

通过命令行传递的参数总是优先。例如，如果在命令行上提供 `--user` 和 `--password` 并且还提供了 `--useDefault` 或 `-z` 参数，那么将使用来自命令行的用户和密码，并且将从 `Authorization.properties` 文件检索主机和端口。或者，如果全都在命令行上提供用户、密码、主机和端口，但是还使用 `--useDefault` 或 `-z` 参数，那么将忽略 `--useDefault` 并且仅使用命令行信息。

对于此处描述的所有 API，语法和示例使用 `-z` 参数来使用最小数量的必需参数。

## 内容存储库脚本编制

内容存储库脚本编制提供能力来处理存储库资源，例如，文件和文件夹。此区域包含以下功能：

- 创建和删除文件夹
- 上传和下载文件
- 导出和导入文件夹
- 管理标签、安全性和元数据

此部分概述存储库功能脚本的 Python 命令行用法。每个操作都包含详细的语法信息、示例和期望的消息。

## 关键字

第 5 页的表 3 列出存储库功能支持的关键字。第二列列出关键字的任何可选短版本。

重要：关键字区分大小写。



表 3. 存储库 API 的关键词.

关键字	可选短版本	用途
--source	-s	源文件或文件夹路径
--target	-t	目标文件夹路径
--version	-v	文件版本
--principal	-r	需要授予许可权的用户
--permission	-n	许可权类型 (例如, 读、写、修改或删除)
--label	-l	分配给文件版本的标签
--criteria	-c	用于搜索文件或文件夹的元数据属性的搜索条件
--author	-a	文件或文件夹的作者名称
--description	-d	文件或文件夹的描述
--title	-i	文件或文件夹的标题
--expirationDate	-q	文件或文件夹的到期日期
--expirationStartDate		文件或文件夹的到期开始日期
--expirationEndDate		文件或文件夹的到期结束日期
--keyword	-k	文件或文件夹的关键词
--cascade	-x	指示文件夹的安全设置应传播到子文件夹和文件
--provider	-f	用于检索主体的安全提供程序
--createVersion	-b	指示要创建的文件的最新版本
--contentLanguage	-g	文件或文件夹的内容语言
--topic		分配给文件或文件夹的主题。您可以输入多个值, 例如, --topic "topic1;topic2"
--modifiedBy		修改文件或文件夹的用户
--mimeType		文件的 MIME 类型
--createdBy		创建文件或文件夹的用户
--submittedHierarchy		指示是否选择已提交作业文件夹
--propertyName		定制属性的名称
--customProperty		要更新的定制属性的"名称/值"对
--propertyName		要检索其有效值的定制属性的名称

对于接受标签和版本信息的所有操作, 用户应指定标签或版本, 但不能同时指定。如果未针对指定的文件指定版本或标签, 那么将使用最新版本。

## 内容存储库操作

### advanceSearch 操作

基于不同的参数搜索存储库中的文件和文件夹。

#### 语法

```
python CADSTool.py advanceSearch --author <author>
--title <title> --description <description>
--createdBy <createdBy> --modifiedBy <modifiedBy>
--keyword <keyword> --label <label>
--topic <topic>
--uri <uri> --parentURI <parentURI>
--expirationStartDate <expirationStartDate>
--expirationEndDate <expirationEndDate>
```

```

--createdStartDate <createdStartDate>
--createdEndDate <createdEndDate>
--objectModifiedStartDate <objectModifiedStartDate>
--objectModifiedEndDate <objectModifiedStartDate>
--versionModifiedStartDate <versionModifiedStartDate>
--versionModifiedEndDate <versionModifiedEndDate>
--submittedHierarchy -z

```

其中：

- <author> 是作者的名称。
- <title> 是文件/文件夹的标题。
- <description> 是文件/文件夹的描述。
- <createdBy> 是创建文件/文件夹的用户的名称。
- <modifiedBy> 是修改文件/文件夹的用户的名称。
- <keyword> 与文件/文件夹相关联的关键字。
- <label> 是版本标记的标签。
- <topic> 是与文件/文件夹相关联的主题。
- <uri> 是文件的 URI。URI 格式为 scpscr:///?id=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx。请注意，URI 搜索中不包含文件夹。
- <parentURI> 是要搜索的文件夹的 URI。URI 格式为 scpscr:///?id=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx。
- <expirationStartDate> 是文件/文件夹的到期开始日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD（例如，1997-07-16T19:20:30+01:00），其中：
  - YYYY = four-digit year
  - MM = 两位数月份（01 是一月，如此类推）
  - DD = 两位数月日期（01 到 31）
  - hh = 两位数小时（00 到 23，无 am/pm）
  - mm = 两位数分钟（00 到 59）
  - ss = 两位数秒（00 到 59）
  - TZD = 时区标志符（Z、+hh:mm 或 -hh:mm）
- <expirationEndDate> 是文件/文件夹的到期结束日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <createdStartDate> 是文件/文件夹的创建开始日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <createdEndDate> 是文件/文件夹的创建结束日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <objectModifiedStartDate> 是文件/文件夹的修改开始日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <objectModifiedEndDate> 是文件/文件夹开始结束日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <versionModifiedStartDate> 是版本的修改开始日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- <versionModifiedEndDate> 是版本的修改结束日期。日期格式为 YYYY-MM-DDThh:mm:ssTZD。
- --submittedHierarchy 指示文件/文件夹位于"已提交作业"文件夹中。

所有参数均为可选。

## 示例

```
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
```

## 消息

在使用此 API 时可能显示以下消息：

- 在 API 成功完成时，将显示匹配搜索条件的所有文件和文件夹的列表。这通常包含文件名及其标准路径和版本。
- Error searching files and folders
- error: no such option:<option>

## ApplySecurity 操作

针对存储库中的文件或文件夹设置安全访问控制表 (ACL) 。

## 语法

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

其中：

- *<source>* 是要应用安全 ACL 的文件或文件夹的标准 IBM SPSS Collaboration and Deployment Services Repository 路径。这是必需参数。
- *<principal>* 是要作为 ACL 的一部分应用于指定的文件或文件夹的用户（例如，*admin*）。这是必需参数。
- *<permission>* 是要应用于指定的文件或文件夹的许可权的类型（例如，读、写、修改、删除或所有者）。这是必需参数。
- *<provider>* 是用于检索有关用户的信息的安全提供程序（主体）。这是可选参数。有效值包括：
  - Native，针对于系统固有的本机本地安全提供程序。这是缺省提供程序。
  - AD\_<name>，针对于 Active Directory，其中 <name> 对应于系统中的安全提供程序名称
  - ADL\_<name>，针对于具有本地替换的 Active Directory，其中 <name> 对应于系统中的安全提供程序名称
  - ldap\_<name>，针对于 OpenLDAP，其中 <name> 对应于系统中的安全提供程序名称
- --cascade，在文件夹上设置安全性时用来将安全性设置传播到指定的文件夹中的所有文件和子文件夹。这是可选参数。

## 示例

以下示例将安全性应用于文件夹：

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

以下示例将安全性应用于文件夹及其所有文件和子文件夹：

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

以下示例将安全应用于名为 SPSSAD 的 Active Directory 安全提供程序中主体的文件夹：

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

以下示例将安全性应用于名为 LDAP 的 OpenLDAP 安全提供程序中的主体文件夹：

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

## 消息

在使用此 API 时可能显示以下消息：

- <permission> permission set successfully for <source>.
- <source> No such file or folder exists. Please try again.
- <permission> Invalid permission type, Please try again.
- <source> Error setting security ACL.

## cascadeSecurity 操作

将文件夹的安全设置传播到文件夹中的所有文件和子文件夹。

## 语法

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

<source> 的值是存储库中文件夹的标准路径。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- Security ACL cascaded successfully for <source>.
- <source> No such folder exists. Please try again.
- <source> Error cascading security ACL.

## CopyResource 操作

将文件或文件夹复制到存储库中的另一个文件夹。

针对此 API 提供重命名功能，可在复制时重命名指定的文件。第 16 页的『moveResource 操作』开头描述的案例也适用于此 copyResource API。

## 语法

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

其中：

- <source> 是要复制的文件/文件夹的标准内容存储库路径。这是必需参数。
- <target> 是要复制的文件/文件夹的标准存储库路径。这是必需参数。

## 示例

以下示例复制文件：

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

以下示例复制并重命名文件：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` copied successfully.
- `<source>` No such file or folder exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error copying file or folder.

## CreateFolder 操作

在存储库中的指定位置创建新文件夹。

### 语法

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

`<source>` 的值是要创建的新文件夹的标准路径。这是必需参数。基于提供的路径，创建新文件夹，包括任何子文件夹。

### 示例

以下示例创建 *Drafts*（如果尚不存在）。

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` Folder created successfully.
- `<source>` No such folder exists. Please try again.
- `<folder>` Folder already exists. Please try again.
- `<source>` Error creating folder.

## deleteFile 操作

删除存储库中的文件，包括其所有版本。

### 语法

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

其中：

- `<source>` 是要删除文件的标准存储库路径。这是必需参数。
- `--submittedHierarchy` 删除"已提交作业"文件夹中的文件。这是可选参数。

### 示例

以下示例删除存储库中的文件，包括其所有版本：

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

以下示例删除"已提交作业"文件夹中的文件，包括其所有版本：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` deleted successfully.
- `<source>` No such file exists. Please try again.
- `<source>` Error deleting file.

## deleteFileVersion 操作

删除存储库中某个文件的一个特定版本。

### 语法

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

其中：

- `<source>` 是要删除文件的标准存储库路径。这是必需参数。
- `<version>` 是要删除的文件的特定版本。这是可选参数。
- `<label>` 是要删除的文件的标签。这是可选参数。
- `--submittedHierarchy` 删除"已提交作业"文件夹中文件的特定版本。这是可选参数。

### 示例

以下示例删除文件的特定版本：

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

以下示例删除具有特定标签的文件：

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

以下示例删除"已提交作业"文件夹中具有特定标签的文件：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` deleted successfully.
- `<source>` No such file exists. Please try again.
- `<source>` Error deleting file.

## deleteFolder 操作

删除存储库中的文件夹，包括其所有内容。

### 语法

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

其中：

- `<source>` 是要删除的文件夹的标准存储库路径。这是必需参数。
- `--submittedHierarchy` 删除"已提交作业"文件夹中文件夹的特定版本。这是可选参数。

## 示例

以下示例删除文件夹：

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

以下示例删除"已提交作业"文件夹中的文件夹：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` deleted successfully.
- `<source>` No such folder exists. Please try again.
- `<source>` Error deleting folder.

## downloadFile 操作

将文件的特定版本从存储库下载到本地文件系统。

## 语法

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

其中：

- `<source>` 是标准存储库路径。这是必需参数。
- `<version>` 是要下载的文件版本。这是可选参数。
- `<label>` 是要下载的文件标签。这是可选参数。
- `<target>` 是要安装文件的标准路径（在本地文件系统上）。

## 示例

以下示例下载文件的最新版本：

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

以下示例使用版本标记下载文件的特定版本：

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

以下示例下载文件的标签版本：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` File downloaded successfully.
- `<source>` No such file exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error downloading File.

## export 操作

从内容存储库启动导出，允许您选择要导出的文件和文件夹，并将 \*.pes 导出文件保存到本地文件系统。

### 语法

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

其中：

- <source> 是要导出的文件夹的标准存储库路径。这是必需参数。
- <target> 是要创建的 \*.pes 导出文件的标准路径（在本地文件系统中）。这是必需参数。

### 示例

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

### 消息

在使用此 API 时可能显示以下消息：

- <source> exported successfully.
- <source> No such folder exists. Please try again.
- <source> Error exporting folder.

## getAccessControlList 操作

检索内容存储库中指定的文件/文件夹的安全访问控制表 (ACL)。

### 语法

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

<source> 的值是文件/文件夹的标准路径。这是必需参数。

### 示例

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

### 消息

在使用此 API 时可能显示以下消息：

- <source> No such file or folder exists. Please try again.
- Error retrieving security details for <source>.

## getAllVersions 操作

检索存储库中文件的所有版本的列表。

### 语法

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

其中：

- <source> 是要检索其版本的文件的标准存储库路径。这是必需参数。
- --submittedHierarchy 从"已提交作业"文件夹检索版本。这是可选参数。



## 示例

以下示例检索指定文件的所有版本：

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

以下示例从"已提交作业"文件夹检索指定文件的所有版本：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file versions.
- 在该过程成功完成时，将显示每个文件版本的信息，包括版本标记和标签信息。

## getChildren 操作

检索指定的存储库文件夹中的所有文件和文件夹的列表。

## 语法

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

`<source>` 的值是文件夹的标准路径。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- 在命令成功完成时，将列出指定的文件夹的所有内容。
- `<source>` No such folder exists. Please try again.
- `<source>` Error getting resources.

## getCustomPropertyValue 操作

检索指定的定制属性接受的有效值。

## 语法

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

`<propertyName>` 的值是定制属性的名称。这是可选参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- `<propertyName>` 将值作为 `<valid values>`
- 检索 `<propertyName>` 的属性详细信息时出错。

## getMetadata 操作

检索存储库中文件或文件夹的元数据属性。

### 语法

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

其中：

- `<source>` 是要检索其元数据的文件或文件夹的标准存储库路径。对于文件夹，将忽略 `version/label` 属性。这是必需参数。
- `<version>` 是要检索其元数据的文件的版本。这是可选参数。
- `<label>` 是要检索其元数据的文件的标签。这是可选参数。
- `--submittedHierarchy` 从"已提交作业"文件夹检索元数据。这是可选参数。

### 示例

以下示例检索文件夹的元数据：

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

以下示例检索文件的标签版本的元数据：

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

以下示例检索"已提交作业"文件夹中文件的标签版本的元数据：

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

### 消息

在使用此 API 时可能显示以下消息：

- `<source>` No such file exists. Please try again.
- `<source>` Error retrieving file metadata.
- 在处理成功完成时，将显示指定文件或文件夹的所有元数据信息，包括任何定制元数据属性。

## import 操作

将现有 \*.pes 导出文件从本地文件系统导入到存储库。

### 语法

```
python CADSTool.py import --source "<source>" --target "<target>"  
--resourceType "<type>"  
--resourceConflict "<rconflict>"  
--labelFrom "<label>"  
--lockResolution "<resolution>"  
--invalidVersionConflict "<vconflict>"  
--resourceDef "<rdefinition>"  
--exclude "<exclusion>"  
-z
```

其中：

- `<source>` 是要导入存储库的 \*.pes 导出文件的标准路径（在本地文件系统上）。这是必需参数。
- `<target>` 是要将 \*.pes 导出文件导入的标准存储库路径。这是必需参数。
- `<type>` 指示正在导入的内容的类型。指定以下一个值：
  - **ContentRepository**，针对于内容对象，例如，文件和文件夹

- **ResourceDef**, 针对于资源定义
- **Credential**, 针对于用户凭证
- **DataSource**, 针对于数据源定义
- **MessageDomain**, 针对于消息域
- **ServerCluster**, 针对于服务器集群定义
- **Server**, 针对于服务器定义
- **PromotionPolicy**, 针对于升级策略

这是可选参数。如果未指定此参数, 那么将使用缺省值 **ContentRepository**。

- **<rconflict>** 指示如何解决重复标识或名称冲突。指定以下一个值:
  - **keepTarget**. 将保留目标项。将忽略 .pes 文件中包含的具有重复标识的源项。
  - **addNewVersion**. 此选项通常用于解决标识冲突或命名冲突。如果在源对象和目标对象之间发生重复标识冲突, 那么将在目标位置创建新版本的对象。如果发生命名冲突, 那么将在目标位置重命名导入的对象。通常, 重命名的对象附加 **\_1**、**\_2**, 如此类推。如果两个版本的对象具有相同标签, 那么系统保留一个标签并丢弃重复的标签, 因为相同项的两个版本不能具有相同标签。保留的标签取决于 **labelFrom** 参数。

这是可选参数。如果未指定此参数, 那么将使用缺省值 **keepTarget**。

- **<label>** 指定在一个对象的两个版本具有相同标签时要使用的标签。将丢弃另一个版本的标签。指定以下一个值:
  - **source**
  - **target**

这是可选参数。如果未指定此参数, 那么将使用缺省值 **source**。

- **<resolution>** 定义在遇到锁定的资源时如何继续。指定以下一个值:
  - **continue**. 继续导入, 忽略任何锁定的资源。
  - **abort**. 如果遇到任何锁定的资源, 终止导入过程。如果由于对象锁定而遇到任何冲突, 那么导入过程将终止并失败。

这是可选参数。如果未指定此参数, 那么将使用缺省值 **abort**。

- **<vconflict>** 定义在导入过程中遇到无效版本时如何继续。指定以下一个值:
  - **import**. 将导入无效版本。
  - **discard**. 将删除无效版本。

这是可选参数。如果未指定此参数, 那么将使用缺省值 **import**。

- **<rdefinition>** 定义资源定义的处理行为。指定以下一个值:
  - **recommended**. 仅当标识或名称与目标定义相冲突时, 才导入资源定义。不会导入具有冲突的任何资源定义。
  - **include**. 从导入文件导入所有资源定义。您可以通过选择相应的复选框, 选择一个或多个要从导入中排除的资源定义类型。
  - **exclude**. 不从导入文件导入资源定义。可能需要修改导入的对象以引用可用资源定义。

这是可选参数。如果未指定此参数, 那么将使用缺省值 **recommended**。

- **<exclusion>** 定义在导入期间排除的资源类型。可以按任意顺序将多个值组合为分号分隔的列表。指定以下一个或多个值:
  - **credential** 排除用户凭证
  - **customproperty** 排除资源对象的定制属性

- **datasource** 排除数据源定义
- **messagedomain** 排除消息域
- **notification** 排除通知定义
- **servercluster** 排除服务器集群定义
- **server** 排除服务器定义
- **topic** 排除主题定义

这是可选参数。如果未指定此参数，那么导入中将包含所有类型。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` imported successfully.
- `<source>` No such file exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error importing folder.

## moveResource 操作

将文件或文件夹移动到存储库中的另一个文件夹。

针对此 API 提供重命名功能，可在移动时重命名指定的文件/文件夹。以下案例描述重命名功能的行为：

如果源为 `/Temp Folder/Temp.txt` 并且目标为 `/Demo Folder`：

- **案例 1**：如果文件夹 `Demo Folder` 存在，将 `Temp.txt` 移至 `Demo Folder`。
- **案例 2**：如果文件夹 `Demo Folder` 不存在，将 `Temp.txt` 移至 `[/]` 并重命名为 `Demo Folder`。

如果源为 `/Temp Folder/Temp.txt` 并且目标为 `/Demo Folder/Abc.dat`：

- **案例 1**：如果文件夹 `Demo Folder` 存在，将 `Temp.txt` 移至 `Demo Folder` 并重命名为 `Abc.dat`。
- **案例 2**：如果文件夹 `Demo Folder` 不存在，将显示错误。

## 语法

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

其中：

- `<source>` 是要移动的文件/文件夹的标准存储库路径。这是必需参数。
- `<target>` 是要移动的文件/文件夹的标准存储库路径。这是必需参数。

## 示例

以下示例移动文件：

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

以下示例移动文件夹：

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

以下示例移动并重命名文件：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` moved successfully.
- `<source>` No such file or folder exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error moving file or folder.

## removeLabel 操作

除去存储库中文件的标签。

## 语法

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

其中：

- `<source>` 是存储库中文件的标准路径。这是必需参数。
- `<label>` 是要从指定的文件除去的标签名称。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- Label removed successfully for `<source>`.
- `<source>` No such folder exists. Please try again.
- `<source>` Error deleting label.
- `<label>` No such label exists. Please try again.

## removeSecurity 操作

从存储库中指定的文件或文件夹除去安全访问控制表 (ACL)。

## 语法

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

其中：

- `<source>` 是要从中除去安全性的文件/文件夹的标准路径。这是必需参数。
- `<principal>` 是要从中除去安全性的指定的文件/文件夹的用户/主体（例如，*admin*）。这是必需参数。
- `<provider>` 是用于检索有关用户的信息的安全提供程序（主体）。这是可选参数。有效值包括：
  - Native, 针对于系统固有的本机本地安全提供程序。这是缺省提供程序。
  - AD\_*<name>*, 针对于 Active Directory, 其中 *<name>* 对应于系统中的安全提供程序名称

- ADL\_<name>, 针对于具有本地替换的 Active Directory, 其中 <name> 对应于系统中的安全提供程序名称
- ldap\_<name>, 针对于 OpenLDAP, 其中 <name> 对应于系统中的安全提供程序名称
- --cascade, 在除去文件夹中的安全性时用于从指定的文件夹中的所有文件和子文件夹除去安全设置。这是可选参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息:

- <source> All the security ACL removed successfully.
- <source> No such folder exists. Please try again.
- <source> Error deleting security ACL.

## search 操作

搜索存储库中的文件和文件夹。结果是匹配搜索条件的文件/文件夹及其版本的列表。

## 语法

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

<criteria> 的值是用于搜索存储库中的所有文件和文件夹的元数据的搜索字符串。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息:

- 在搜索成功完成时, 将显示匹配搜索条件的所有文件和文件夹的列表。这通常包含文件名及其标准路径和版本。
- <criteria> No file or folder matches the search criteria.
- Error searching files and folders.

## setLabel 操作

将标签应用于存储库中文件的版本。如果文件已添加标签, 那么将除去原始标签并替换为新标签。

## 语法

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

其中:

- <source> 是存储库中文件的标准路径。这是必需参数。
- <version> 是要应用标签的文件的版本。这是必需参数。
- <label> 是要应用于指定的文件版本的标签名称。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- Label set successfully for <source>.
- <source> No such folder exists. Please try again.
- <source> Error setting label.

## setMetadata 操作

将元数据属性应用于存储库中的文件和文件夹。

表 4 列出元数据属性及其是否可应用于文件和文件夹。

表 4. 元数据属性和资源类型.

元数据属性	资源类型
作者	文件
描述	文件或文件夹
标题	文件或文件夹
到期日期	文件或文件夹
关键字	文件
主题	文件
定制元数据	文件或文件夹

## 语法

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

其中：

- <source> 是要在其中设置元数据的文件或文件夹的标准存储库路径。这是必需参数。
- <author> 是文件或文件夹的作者。这是可选参数。
- <title> 是文件或文件夹的标题。这是可选参数。
- <description> 是文件/文件夹的描述。这是可选参数。
- <expirationDate> 是文件或文件夹的到期日期。这是可选参数。日期格式为 YYYY-MM-DDThh:mm:ssTZD（例如，1997-07-16T19:20:30+01:00），其中：

YYYY = four-digit year

MM = 两位数月份（01 是一月，如此类推）

DD = 两位数月日期（01 到 31）

hh = 两位数小时（00 到 23，无 am/pm）

mm = 两位数分钟（00 到 59）

ss = 两位数秒 (00 到 59)

TZD = 时区标志符 (Z、+hh:mm 或 -hh:mm)

- `<keyword>` 是文件或文件夹的关键字。这是可选参数。
- `<version>` 是要应用元数据的文件的版本。这是可选参数。
- `<label>` 是要在其中应用元数据的文件的标签版本。这是可选参数。
- `<topic>` 是要应用于文件或文件夹的主题。这是可选参数。
- `<customProperty>` 是要应用于文件或文件夹的定制属性值。这是可选参数。该值指定为 `<customProperty>=<value>`。要应用多个定制属性，请使用分号 ( ;) 作为分隔符 (`<customProperty>=<value>;<customProperty>=<value>`)。用 `|` 运算符 (`<customProperty>=opt1|opt2;<customProperty>=value`) 分隔多选属性值。

注：必须至少提供一个可选参数才能使用 setMetadata API。

## 示例

```
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
```

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` Metadata set successfully.
- `<source>` No such file or folder exists. Please try again.
- `<source>` Error setting metadata.

## uploadFile 操作

使用如果已存在则创建文件的新版本的选项，将文件从本地文件系统保存到内容存储库。

## 语法

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

其中：

- `<source>` 是要上载的文件的标准路径（在本地文件系统中）。这是必需参数。
- `<target>` 是要上载文件的存储库中文件夹的标准路径。这是必需参数。
- `--createVersion` 指示指定的文件已存在，并且应创建新版本。这是可选参数。

## 示例

在以下示例中，目标是 *Drafts* 的标准路径：

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

如果 */Demo/Drafts* 文件夹中已存在 *MyReport.rptdesign*，那么使用 `--createVersion` 参数：

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

## 消息

在使用此 API 时可能显示以下消息：

- `<source>` File uploaded successfully.



- `<source>` No such file exists. Please try again.
- `<target>` No such folder exists. Please try again.
- `<source>` Error Uploading File.

## 流程管理功能

流程管理脚本编制提供能力来处理作业。此区域包含以下功能：

- 执行作业
- 检索作业历史记录
- 检索作业详细信息

此部分概述流程管理功能脚本的 Python 命令行用法。每个 API 都包含详细的语法信息、示例和预期消息内容。

## 关键字

表 5 列出“流程管理”API 支持的关键字。第二列列出提供的关键字的任何可选短版本。表仅列出特定于“流程管理”API 的关键字。有关应用于“流程管理”API 和存储库 API 的其他关键字，请参阅第 3 页的表 2 和第 5 页的表 3。

表 5. “流程管理”API 的关键字

关键字	可选短版本	用途
<code>--source</code>	<code>-s</code>	源作业，包括路径
<code>--target</code>	<code>-t</code>	目标文件夹路径
<code>--notification</code>	<code>-j</code>	指示作业将含通知运行
<code>--async</code>	<code>-m</code>	指示作业将异步运行
<code>--execId</code>	<code>-y</code>	作业的执行标识
<code>--jobStepName</code>	<code>-q</code>	作业步骤名称
<code>--log</code>		指示不应删除日志。如果与 <code>--target</code> 结合使用，那么日志将存储在 <code>--target</code> 指定的位置。否则，将内嵌显示日志。

## 流程管理操作

### deleteJobExecutions 操作

删除指定的祖业执行对象。

#### 语法

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

`<execIDs>` 的值是要删除的执行的空格分隔的标识列表。这是必需参数。

#### 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- Execution Id not specified.

## executeJob 操作

根据传递的参数并行或异步运行作业。在同步运行的情况下，API 直至作业完成才返回。在异步运行的情况下，API 在作业启动后返回。

## 语法

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

其中：

- <source> 是存储库中作业的标准路径。这是必需参数。
- --notification 用于运行含通知的作业。这是可选参数。
- --async 用于异步运行作业。这是可选参数。

## 示例

以下示例同步运行不含通知的作业：

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

以下示例同步运行含通知的作业：

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

以下示例异步运行不含通知的作业：

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

以下示例异步运行含通知的作业：

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

## 消息

在使用此 API 时可能显示以下消息：

- <source> Job executed successfully. Job execution Id is <execId>.
- <source> No such job exists. Please try again.
- <source> Error executing job.

## getJobExecutionDetails 操作

列出某一特定作业的运行详细信息，其中包括所有作业步骤和迭代。

## 语法

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

其中：

- <execId> 是作业的执行标识。这是必需参数。
- --log 指示应内嵌显示的作业日志。如果未包含 --log 参数，那么不会显示作业步骤运行所生成的任何日志。这是可选参数。
- <target> 是用于存储日志的位置（在本地文件系统上）。这是可选参数，并且仅与 --log 参数结合使用。

## 示例

以下示例列出特定作业运行的详细信息：

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

以下示例列出特定日志运行的详细信息，并且内嵌显示日志：

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

以下示例列出特定作业运行的详细信息，并且作业日志存储在特定位置：

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

## 消息

在使用此 API 时可能显示以下消息：

- 如果成功运行，那么将针对作业、作业步骤和作业迭代列出所有运行详细信息。日志内嵌显示或者保存到本地文件系统上指定的位置。
- `<execId>` No such execution exists. Please try again.
- `<execId>` Error displaying details of a job execution.
- `--target` cannot be used without `--log` parameter

## getJobExecutionList 操作

针对作业的所有版本，列出特定作业的当前运行和已完成的运行。

## 语法

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

`<source>` 的值是存储库中作业的标准路径。这是必需参数。

## 示例

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

## 消息

在使用此 API 时可能显示以下消息：

- 如果指定的作业成功运行，那么将列出所有运行详细信息，例如，执行标识、作业名称、作业执行状态以及作业执行开始和结束时间。
- `<source>` No such job exists. Please try again.
- `<source>` Error displaying execution list for a job.



---

## 第 3 章 PESImpl 模块

IBM SPSS Collaboration and Deployment Services - Essentials for Python 工具允许直接在 Python 脚本中与 IBM SPSS Collaboration and Deployment Services Repository 进行对象。

在 Python 代码中，从 `pes.api.PESImpl` 模块导入 `PESImpl` 类。创建使用要连接到的存储库的连接信息的 `PESImpl` 对象。

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

`PESImpl` 构造方法的参数如下所示：

- `user` 对应于用户的名称。如果用户不是来自于本机提供程序，那么值应包含表示安全提供程序的前缀。以下前缀值有效：
  - `Native`，针对于系统固有的本机本地安全提供程序。这是缺省提供程序。
  - `AD_<name>`，针对于 Active Directory，其中 `<name>` 对应于系统中的安全提供程序名称
  - `ADL_<name>`，针对于具有本地替换的 Active Directory，其中 `<name>` 对应于系统中的安全提供程序名称
  - `ldap_<name>`，针对于 OpenLDAP，其中 `<name>` 对应于系统中的安全提供程序名称

前缀后跟斜杠和用户名。对于 Active Directory 提供程序，前缀中包含域。例如，对于 Active Directory 实例 `AD_SPSSAD` 的 `MYDOMAIN` 域中的用户 `icrod`，值为 `AD_SPSSAD/MYDOMAIN/icrod`。如果用户 `icrod` 位于 OpenLDAP 提供程序 `SPSSLDAP` 中，那么值为 `ldap_SPSSLDAP/icrod`。

- `password` 对应于与指定的用户相关联的密码
- `host` 指定存储库服务器的名称
- `port` 指定存储库服务器的端口号
- `ssl=True` 指示存储库服务器使用安全套接字层 (SSL) 协议来加密通信。如果 `ssl` 参数设置为 `False`，或者如果在创建 `PESImpl` 对象时省略参数，那么服务器通信将不使用 SSL。在使用 SSL 时，必须针对 SSL 配置存储库服务器。有关更多信息，请参阅管理员文档。

或者，您可以指定服务器 URL，代替 `host`、`port` 和 `ssl` 参数。

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("user", "password", server_url="url")
```

`server_url` 参数指定存储库服务器的完整 URL。如果服务器使用定制上下文路径，那么使用此参数。例如，以下构造方法对应于名为 `myserver` 的服务器，其在端口 443 上使用 SSL 并且上下文路径为 `/ibm/spss`：

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

注：IPv6 地址必须用方括号括起来了，例如，`[3ffe:2a00:100:7031::1]`。

然后，可以使用 `pesImpl` 对象来访问特定方法。

---

## 内容存储库 API

内容存储库脚本编制提供能力来处理存储库资源，例如，文件和文件夹。此区域包含以下功能：

- 创建和删除文件夹
- 上传和下载文件
- 导出和导入文件夹
- 管理标签、安全性和元数据

此部分概述用于处理存储库中存储的资源的 PESImpl API。每个方法都包含详细的语法信息、示例和期望的消息。

## 方法

以下部分列出 IBM SPSS Collaboration and Deployment Services 支持的所有内容存储库方法。

注：

- 对于具有可选参数 Label 和 Version 的所有方法，使用 Label 或 Version，但不能同时使用。如果未针对指定的文件或文件夹指定 Version 或 Label，那么将使用最新版本。
- 对于需要存储库中文件或文件夹路径的所有方法，可以使用路径或对象 URI。可以通过在 IBM SPSS Deployment Manager 中查看对象属性来获取对象 URI。
- 有关需要输入包含非拉丁语 Unicode 字符的源或目标存储库或文件系统路径的方法，字符串必须指定为 Unicode 对象，例如：

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

## advanceSearch 方法

基于不同传递的参数作为参数来搜索存储库中的文件和文件夹。

您可以在以下项上执行搜索：

- 作者
- 描述
- 标题
- 创建者
- 修改者
- 到期开始日期
- 到期结束日期
- MIME 类型
- 标签
- 关键字
- 主题
- 创建开始日期
- 创建结束日期
- 版本修改开始日期
- 对象修改结束日期
- 对象修改开始日期

- 版本修改开始日期
- 版本修改结束日期
- 父文件夹 URI
- 资源 URI

`advanceSearch(criteriaDict,submittedHierarchy)`

表 6. `advanceSearch` 的输入参数.

字段	使用	类型	描述	示例值
<code>criteriaDict</code>	必需	字典	字典包含可针对其执行搜索的键/值对。可接受的键值为： <ul style="list-style-type: none"> <li>• author</li> <li>• title</li> <li>• description</li> <li>• createdBy</li> <li>• modifiedBy</li> <li>• expirationStartDate</li> <li>• expirationEndDate</li> <li>• mimeType</li> <li>• label</li> <li>• keyword</li> <li>• topic</li> <li>• createdStartDate</li> <li>• createdEndDate</li> <li>• objectModifiedStartDate</li> <li>• objectModifiedEndDate</li> <li>• versionModifiedStartDate</li> <li>• versionModifiedEndDate</li> <li>• parentURI</li> <li>• uri</li> </ul>	{ "author":"admin", "title":"search", "label":"label 1"}
<code>submittedHierarchy</code>	可选	布尔值	指示是否选择已提交作业文件夹	True 或 False

请注意，当前 `expirationStartDate` 和 `expirationEndDate` 在与其他搜索字段（例如，标题或作者）一起使用时，将不起作用。

表 7. `advanceSearch` 的返回值.

类型	描述
<code>PageResult</code>	每个行对应于一个搜索匹配的结构。请参阅主题第 49 页的『 <code>PageResult</code> 类』以获取更多信息。

表 8. `advanceSearch` 的异常.

类型	描述
<code>InsufficientParameterException</code>	未指定必填参数。

### 示例：按标签和关键字执行搜索

以下样本返回关键字值为 `Quarterly` 的标注为 `Production` 的所有文件版本。

```

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()

```

## 示例：按 URI 搜索

以下样本返回具有指定的 URI 的文件的所有文件版本。

```

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()

```

## applySecurity 方法

针对存储库中的文件或文件夹设置安全访问控制表 (ACL)。

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

表 9. *applySecurity* 的输入参数.

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<i>principal</i>	必需	字符串	要作为 ACL 的一部分应用于指定的文件或文件夹的用户（例如， <i>admin</i> ）	admin
<i>permission</i>	必需	字符串	应用于指定的文件或文件夹的许可权的类型	READ, WRITE, DELETE, MODIFY_ACL, OR OWNER



表 9. *applySecurity* 的输入参数 (续).

字段	使用	类型	描述	示例值
<i>provider</i>	可选	字符串	用于将安全性应用于用户 (例如, <i>Native</i> ) 的安全提供程序。有效值包括: <ul style="list-style-type: none"> <li>• <i>Native</i>, 针对于系统固有的本机本地安全提供程序。这是缺省提供程序。</li> <li>• <i>AD_&lt;name&gt;</i>, 针对于 Active Directory, 其中 <i>&lt;name&gt;</i> 对应于系统中的安全提供程序名称</li> <li>• <i>ADL_&lt;name&gt;</i>, 针对于具有本地替换的 Active Directory, 其中 <i>&lt;name&gt;</i> 对应于系统中的安全提供程序名称</li> <li>• <i>ldap_&lt;name&gt;</i>, 针对于 OpenLDAP, 其中 <i>&lt;name&gt;</i> 对应于系统中的安全提供程序名称</li> </ul>	<i>Native</i>
<i>cascade</i>	可选	布尔值	将安全设置传播到指定的文件夹中的所有文件和子文件夹	True 或 False

表 10. *applySecurity* 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 11. *applySecurity* 的异常.

类型	描述
<i>ResourceNotFoundException</i>	源文件不存在。
<i>InsufficientParameterException</i>	未指定必需参数。
<i>IllegalParameterException</i>	指定的用户或安全提供程序名称不正确。

## 示例

以下样本将指定的文件的 *READ* 许可权分配给用户。

```

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

## **cascadeSecurity** 方法

将文件夹的安全设置传播到文件夹中的所有文件和子文件夹。

```
cascadeSecurity(source)
```

表 12. *cascadeSecurity* 的输入参数。

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	存储库中文件夹的标准路径或对象 URI	"/Temp Folder" 或 "0a58c3670016a7860000010dcee0eaa28219"

表 13. *cascadeSecurity* 的返回值。

类型	描述
Boolean	True 或 False，基于方法是否成功运行。

表 14. *cascadeSecurity* 的异常。

类型	描述
ResourceNotFoundException	源文件夹不存在。
InsufficientParameterException	未指定必需参数。

## 示例

以下样本将文件 *Projects* 的安全性层叠到文件夹的内容。

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

## copyResource 方法

将文件或文件夹复制到存储库中的另一个文件夹。在复制时可重命名指定的源文件或文件夹。

请参阅第 41 页的『*moveResource* 方法』以获取有关重命名的更多信息。

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

表 15. *copyResource* 的输入参数。

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<i>target</i>	必需	字符串	要将文件复制到的文件夹的标准路径或对象 URI。在复制时，还可以提供新文件名来重命名指定的文件或文件夹。	"/New Folder" 或 "/New Folder/abc.dat"

表 16. *copyResource* 的返回值。

类型	描述
String	复制的文件或文件夹的 URI

表 17. *copyResource* 的异常。

类型	描述
ResourceNotFoundException	源文件或目标文件夹不存在。

表 17. `copyResource` 的异常 (续).

类型	描述
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下样本将 `Drafts` 文件夹复制到名为 `Projects` 的文件夹。

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

## createFolder 方法

在存储库中的指定位置创建新文件夹。

```
createFolder(source)
```

表 18. `createFolder` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	要在存储库中创建的文件夹	/Temp Folder/Sample Folder

表 19. `createFolder` 的返回值.

类型	描述
<code>String</code>	创建的文件夹的 URI

表 20. `createFolder` 的异常.

类型	描述
<code>InsufficientParameterException</code>	未指定必需参数。
<code>ResourceAlreadyExistsException</code>	存储库中已存在指定的文件夹。

## 示例

以下示例创建名为 `Drafts` 的文件夹作为 `Demo` 文件夹的子代。如果创建文件夹时发生问题，那么会向控制台发送异常消息。

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

## deleteFile 方法

删除存储库中的文件。将删除文件的所有版本。

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

表 21. `deleteFile` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	可选	布尔值	指示文件是否位于已提交作业文件夹中	True 或 False

表 22. `deleteFile` 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 23. `deleteFile` 的异常.

类型	描述
ResourceNotFoundException	源文件不存在。
InsufficientParameterException	未指定必需参数。
IllegalParameterException	要删除的指定资源是文件夹。

## 示例

以下示例删除存储库中的文件 `MyReport.rptdesign`。

```
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."
```

## deleteFileVersion 方法

删除存储库中某个文件的一个特定版本。

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

表 24. `deleteFileVersion` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>version</code>	可选。但是，必须指定 <code>version</code> 或 <code>label</code> 。	字符串	要删除的文件的特定版本	"0:2006-08-25 21:15:49.453"

表 24. `deleteFileVersion` 的输入参数 (续).

字段	使用	类型	描述	示例值
<code>label</code>	可选。但是，必须指定 <code>version</code> 或 <code>label</code> 。	字符串	要删除的文件的特别标记版本	"Version 1"
<code>submittedHierarchy</code>	可选	布尔值	指示文件是否位于已提交作业文件夹中	True 或 False

表 25. `deleteFileVersion` 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 26. `deleteFileVersion` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件或目标文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。
<code>IllegalParameterException</code>	要删除的指定资源是文件夹。

## 示例

以下示例从存储库中删除标签为 `Test` 的文件 `MyReport.rptdesign` 的版本。

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

## `deleteFolder` 方法

删除存储库中的文件夹及其内容。

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

表 27. `deleteFolder` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件夹的标准路径或对象 URI	"/Temp Folder" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	可选	布尔值	指示文件夹是否位于已提交作业文件夹中	True 或 False

表 28. `deleteFolder` 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 29. `deleteFolder` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	指定的文件夹不存在。

表 29. `deleteFolder` 的异常 (续).

类型	描述
<code>InsufficientParameterException</code>	未指定必需参数。
<code>IllegalParameterException</code>	要删除的指定资源不是文件夹。

## 示例

以下示例删除存储库中名为 `Drafts` 的文件夹。如果删除文件夹时发生问题，那么会向控制台发送异常消息。

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

## downloadFile 方法

将文件的特定版本从存储库下载到本地文件系统。

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

表 30. `downloadFile` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	要下载的文件的标准存储库路径	<code>"/Temp Folder/Temp.txt"</code>
<code>target</code>	必需	字符串	要将文件下载到的文件夹的标准路径 (在本地文件系统上)	<code>"C:\Temp"</code>
<code>version</code>	可选。可指定版本或标签。	字符串	要下载的文件的标准版本	<code>"0:2006-08-25 21:15:49.453"</code>
<code>label</code>	可选。可指定版本或标签。	字符串	要下载的文件的标准标签版本	<code>"Version 2"</code>

表 31. `downloadFile` 的返回值.

类型	描述
<code>Resource</code>	有关存储库对象的信息的容器。请参阅主题第 48 页的『Resource 类』以获取更多信息。

表 32. `downloadFile` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件或目标文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下样本将标签为 `Production` 的文件 `MyReport.rptdesign` 版本下载到本地文件系统上的 `Shared` 目录。

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

## exportResource 方法

将指定的存储库文件夹导出到本地文件系统中指定的 \*.pes 导出文件。

`exportResource(source, target)`

表 33. `exportResource` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	要导出的文件夹的标准存储库路径或对象 URI	/Temp Folder 或 0 a 5 8 c 3 6 7 0 0 1 6 a 7 8 6 0 0 0 0 0 1 0 d c e e 0 e a a 2 8 2 1 9
<code>target</code>	必需	字符串	要将文件夹导出到的标准路径（在本地文件系统中）和文件名	C:\Temp\backup.pes

表 34. `exportResource` 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 35. `exportResource` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件或目标文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。
<code>IllegalParameterException</code>	指定的目标是文件夹。目标必须是 *.pes 文件。

## 示例

以下样本将 `Drafts` 文件夹的内容导出到本地文件系统中 `backups` 文件夹中的导出文件。

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

## getAccessControlList 方法

针对存储库中指定的文件或文件夹检索安全访问控制表 (ACL)。

`getAccessControlList(source, submittedHierarchy)`

表 36. `getAccessControlList` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 " 0 a 5 8 c 3 6 7 0 0 1 6 a 7 8 6 0 0 0 0 0 1 0 d c e e 0 e a a 2 8 2 1 9 "
<code>submittedHierarchy</code>	可选	布尔值	指示文件或文件夹是否位于已提交作业文件夹中	True 或 False

表 37. `getAccessControlList` 的返回值.

类型	描述
Dictionary	显示包含用户名和关联的许可权的字典。例如： { "admin": "MODIFY_ACL", "Joe": "DELETE" }

表 38. `getAccessControlList` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件或目标文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下示例打印文件 `MyReport.rptdesign` 的 ACL。

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

## getAllVersions 方法

检索存储库中文件的所有版本的列表。

`getAllVersions(source, submittedHierarchy)`

表 39. `getAllVersions` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	可选	布尔值	指示文件是否位于已提交作业文件夹中	True 或 False

表 40. `getAllVersions` 的返回值.

类型	描述
List	资源对象的列表。请参阅第 48 页的『Resource 类』。

表 41. `getAllVersions` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件不存在。
<code>InsufficientParameterException</code>	未指定必需参数。
<code>IllegalParameterException</code>	指定的源是文件夹。

## 示例

此示例检索雨哦关站点 `MyReport.rptdesign` 的所有版本的信息，并打印每个的作者、版本标记和版本标签。

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



## getChildren 方法

检索指定的存储库文件夹中所有文件和文件夹的列表。

`getChildren(source,submittedHierarchy)`

表 42. `getChildren` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件夹的标准路径或对象 URI	"/Temp Folder" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>submittedHierarchy</code>	可选	布尔值	指示文件夹是否位于"已提交作业"文件夹中	True 或 False

表 43. `getChildren` 的返回值.

类型	描述
List	资源对象的列表。请参阅第 48 页的『Resource 类』。

表 44. `getChildren` 的异常.

类型	描述
<code>InsufficientParameterException</code>	未指定必需参数。
<code>ResourceNotFoundException</code>	文件夹不存在。

## 示例

以下样本检索 `/Demo/Drafts` 文件夹的内容，打印每个的标题、作者和资源标识。

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

## getCustomPropertyValue 方法

检索指定的定制属性接受的有效值。

`getCustomPropertyValue(propertyName)`

表 45. `getCustomPropertyValue` 的输入参数.

字段	使用	类型	描述	示例值
<code>propertyName</code>	必需	字符串	定制属性的名称	"FreeForm"

表 46. `getCustomPropertyValue` 的返回值.

类型	描述
List	返回定制属性接受的有效值列表。如果需要选择属性（例如，单选或多选），那么列表包含所选内容的所有有效值。如果这是自由格式属性，那么列表包含属性接受的数据类型（例如，String、Date 或 Number）。

表 47. `getCustomPropertyValue` 的异常。

类型	描述
<code>ResourceNotFoundException</code>	指定的属性不存在。
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下样本访问定制属性 `Language` 的值。

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

## getMetadata 方法

检索存储库中文件或文件夹的元数据属性，包括任何定制属性和主题信息。

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

表 48. `getMetadata` 的输入参数。

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>version</code>	可选。可指定版本或标签。	字符串	文件或文件夹的特定版本	"0:2006-08-25 21:15:49.453"
<code>label</code>	可选。可指定版本或标签。	字符串	文件或文件夹的特定标签版本	"Version 1"
<code>submittedHierarchy</code>	可选	布尔值	指示文件是否位于已提交作业文件夹中	True 或 False

表 49. `getMetadata` 的返回值。

类型	描述
<code>Resource</code>	有关存储库对象的信息的容器。请参阅主题第 48 页的『Resource 类』以获取更多信息。

表 50. `getMetadata` 的异常。

类型	描述
<code>ResourceNotFoundException</code>	源文件或文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下示例访问 `/Demo/Drafts` 文件夹的资源标识。

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

## importResource 方法

将现有 \*.pes 导出文件从本地文件系统导入到存储库。

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

表 51. importResource 的输入参数.

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	要导入的文件的的标准路径（在本地文件系统上）	"C:\Temp\New.pes"
<i>target</i>	必需	字符串	要导入到的文件夹的标准存储库路径或对象 URI	"/Temp Folder" 或 "0a58c3670016a7860000010dcee0eaa28219"
<i>resourceType</i>	可选	字符串	要导入的内容类型。指定以下一个值： <ul style="list-style-type: none"> <li>• <b>ContentRepository</b>，针对于内容对象，例如，文件和文件夹</li> <li>• <b>ResourceDef</b>，针对于资源定义</li> <li>• <b>Credential</b>，针对于用户凭证</li> <li>• <b>DataSource</b>，针对于数据源定义</li> <li>• <b>MessageDomain</b>，针对于消息域</li> <li>• <b>ServerCluster</b>，针对于服务器集群定义</li> <li>• <b>Server</b>，针对于服务器定义</li> <li>• <b>PromotionPolicy</b>，针对于升级策略</li> </ul> 如果未指定此参数，那么将使用缺省值 <b>ContentRepository</b> 。	
<i>resourceConflict</i>	可选	字符串	指示如何解决重复标识或名称冲突。指定以下一个值： <ul style="list-style-type: none"> <li>• <b>keepTarget</b>。将保留目标项。将忽略 .pes 文件中包含的具有重复标识的源项。</li> <li>• <b>addNewVersion</b>。此选项通常用于解决标识冲突或命名冲突。如果在源对象和目标对象之间发生重复标识冲突，那么将在目标位置创建新版本的对象。如果发生命名冲突，那么将在目标位置重命名导入的对象。通常，重命名的对象附加 _1、_2，如此类推。如果两个版本的对象具有相同标签，那么系统保留一个标签并丢弃重复的标签，因为相同项的两个版本不能具有相同标签。保留的标签取决于 <b>labelFrom</b> 参数。</li> </ul> 如果未指定此参数，那么将使用缺省值 <b>keepTarget</b> 。	

表 51. *importResource* 的输入参数 (续).

字段	使用	类型	描述	示例值
<i>labelFrom</i>	可选	字符串	在对象的两个版本具有相同标签时要使用的标签。将丢弃另一个版本的标签。指定 <b>source</b> 或 <b>target</b> 。如果未指定此参数，那么将使用缺省值 <b>source</b> 。	
<i>lockResolution</i>	可选	字符串	定义在遇到锁定的资源时如何继续。指定以下一个值： <ul style="list-style-type: none"> <li>• <b>continue</b>。继续导入，忽略任何锁定的资源。</li> <li>• <b>abort</b>。如果遇到任何锁定的资源，终止导入过程。如果由于对象锁定而遇到任何冲突，那么导入过程将终止并失败。</li> </ul> 如果未指定此参数，那么将使用缺省值 <b>abort</b> 。	
<i>invalidVersionConflict</i>	可选	字符串	定义在导入过程中遇到无效版本时如何继续。指定以下一个值： <ul style="list-style-type: none"> <li>• <b>import</b>。将导入无效版本。</li> <li>• <b>discard</b>。将删除无效版本。</li> </ul> 如果未指定此参数，那么将使用缺省值 <b>import</b> 。	
<i>resourceDef</i>	可选	字符串	定义资源定义的处理行为。指定以下一个值： <ul style="list-style-type: none"> <li>• <b>recommended</b>。仅当标识或名称与目标定义相冲突时，才导入资源定义。不会导入具有冲突的任何资源定义。</li> <li>• <b>include</b>。从导入文件导入所有资源定义。您可以通过选择相应的复选框，选择一个或多个要从导入中排除的资源定义类型。</li> <li>• <b>exclude</b>。不从导入文件导入资源定义。可能需要修改导入的对象以引用可用资源定义。</li> </ul> 如果未指定此参数，那么将使用缺省值 <b>recommended</b> 。	

表 51. *importResource* 的输入参数 (续).

字段	使用	类型	描述	示例值
<i>exclude</i>	可选	字符串	定义在导入期间排除的资源类型。可以按任意顺序将多个值组合为分号分隔的列表。指定以下一个或多个值： <ul style="list-style-type: none"> <li>• <b>credential</b> 排除用户凭证</li> <li>• <b>customproperty</b> 排除资源对象的定制属性</li> <li>• <b>datasource</b> 排除数据源定义</li> <li>• <b>messagedomain</b> 排除消息域</li> <li>• <b>notification</b> 排除通知定义</li> <li>• <b>servercluster</b> 排除服务器集群定义</li> <li>• <b>server</b> 排除服务器定义</li> <li>• <b>topic</b> 排除主题定义</li> </ul> 如果未指定此参数，那么导入中将包含所有类型。	

表 52. *importResource* 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

表 53. *importResource* 的异常.

类型	描述
ResourceNotFoundException	源文件或目标文件夹不存在。
InsufficientParameterException	未指定必需参数。

## 示例

以下样本将 *drafts.pes* 导出文件的内容导入到 IBM SPSS Collaboration and Deployment Services Repository 的 */Demo/Drafts* 文件夹

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

## moveResource 方法

将文件或文件夹移动到存储库中的另一个文件夹。指定的源文件在移动时可进行重命名，其最终名称由目标类型和是否存来确定。

下表描述在移动文件时重命名功能部件的行为：

表 54. 文件重命名.

目标类型	目标文件夹存在	目标文件夹不存在
文件夹	源文件称为目标文件夹的子代。	源文件移至指定的目标的父文件夹并重命名为目标文件夹名称。
文件	源文件移至包含目标文件的文件夹，并重命名为目标名称。	报告错误。

例如，如果源位于文件 `/Temp Folder/Temp.txt` 中并且指定的目标为文件夹 `/Demo Folder`，那么可能发生以下结果：

- 如果文件夹 `Demo Folder` 存在，那么 `Temp.txt` 移至 `Demo Folder`。
- 如果文件夹 `Demo Folder` 不存在，那么 `Temp.txt` 移至 `[/]` 并重命名为 `Demo Folder`。

或者，如果源为 `/Temp Folder/Temp.txt` 并且指定的目标为文件 `/Demo Folder/Abc.dat`，那么可能发生以下结果：

- 如果文件夹 `Demo Folder` 存在，那么 `Temp.txt` 移至 `Demo Folder` 并重命名为 `Abc.dat`。
- 如果文件 `Demo Folder` 不存在，那么显示错误。

`moveResource(source, target)`

表 55. `moveResource` 的输入参数。

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	<code>/Temp Folder/Temp.txt</code> 或 <code>0a58c3670016a7860000010dcee0eaa28219</code>
<code>target</code>	必需	字符串	要将文件移动到的文件夹的标准路径或对象 URI。在移动时，还可以提供新文件名来重命名指定的文件或文件夹。	<code>/New Folder</code> 或 <code>/New Folder/abc.dat</code>

表 56. `moveResource` 的返回值。

类型	描述
Boolean	True 或 False，基于方法是否成功运行。

表 57. `moveResource` 的异常。

类型	描述
<code>ResourceNotFoundException</code>	指定的源不存在。
<code>InsufficientParameterException</code>	未指定必需参数。

## 示例

以下样本将文件 `MyReport.rptdesign` 从 `/Demo/Drafts` 文件夹移至 `/Approved` 文件夹。

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

## removeLabel 方法

除去存储库中文件的标签。

`removeLabel(source, label)`

表 58. `removeLabel` 的输入参数。

字段	使用	类型	示例值	描述
<code>source</code>	必需	字符串	<code>"/Temp Folder/Temp.txt"</code> 或 <code>"0a58c3670016a7860000010dcee0eaa28219"</code>	存储库中文件的标准路径或对象 URI

表 58. `removeLabel` 的输入参数 (续).

字段	使用	类型	示例值	描述
<code>label</code>	必需	字符串	"Version 1"	要除去的标签名称

表 59. `removeLabel` 的返回值.

类型	描述
String	更新的文件的 URI

表 60. `removeLabel` 的异常.

类型	描述
ResourceNotFoundException	源文件不存在。
InsufficientParameterException	未指定必需参数。

## 示例

以下样本从文件 `MyReport.rptdesign` 中除去标签 `Draft`。

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

## removeSecurity 方法

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

从存储库中指定的文件或文件夹除去安全访问控制表 (ACL)。

表 61. `removeSecurity` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>principal</code>	必需	字符串	要从指定的文件或文件夹除去的用户 (例如, <code>admin</code> )	admin

表 61. `removeSecurity` 的输入参数 (续).

字段	使用	类型	描述	示例值
<code>provider</code>	可选	字符串	用于获取有关用户的信息的安全提供程序 (例如, <code>Native</code> )。以下值有效: <ul style="list-style-type: none"> <li><code>Native</code>, 针对于系统固有的本机本地安全提供程序。这是缺省提供程序。</li> <li><code>AD_&lt;name&gt;</code>, 针对于 Active Directory, 其中 <code>&lt;name&gt;</code> 对应于系统中的安全提供程序名称</li> <li><code>ADL_&lt;name&gt;</code>, 针对于具有本地替换的 Active Directory, 其中 <code>&lt;name&gt;</code> 对应于系统中的安全提供程序名称</li> <li><code>ldap_&lt;name&gt;</code>, 针对于 OpenLDAP, 其中 <code>&lt;name&gt;</code> 对应于系统中的安全提供程序名称</li> </ul>	<code>Native</code>
<code>cascade</code>	可选	布尔值	将安全设置传播到指定的文件夹中的所有文件和子文件夹	<code>True</code> 或 <code>False</code>

表 62. `removeSecurity` 的返回值.

类型	描述
<code>Boolean</code>	<code>True</code> 或 <code>False</code> , 基于方法是否成功运行。

表 63. `removeSecurity` 的异常.

类型	描述
<code>ResourceNotFoundException</code>	源文件或目标文件夹不存在。
<code>InsufficientParameterException</code>	未指定必需参数。
<code>IllegalParameterException</code>	指定的用户或安全提供程序名称不正确。

## 示例

以下样本从文件 `MyReport.rptdesign` 中除去主体的 ACL。

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

## search 方法

搜索存储库中的文件, 返回具有匹配搜索条件的元数据内容的文件版本的列表。

```
search(criteria)
```

表 64. `search` 的输入参数.

字段	使用	类型	描述	示例值
<code>criteria</code>	必需	字符串	用于搜索文件元数据的值	<code>"Age"</code>



表 65. `search` 的返回值.

类型	描述
PageResult	每个行对应于一个搜索匹配的结构。请参阅主题第 49 页的『PageResult 类』以获取更多信息。

表 66. `search` 的异常.

类型	描述
InsufficientParameterException	未指定必需参数。

## 示例

以下样本搜索在任何元数据字段中包含文本 *Quarterly* 的文件版本。

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

## setLabel 方法

将标签应用于存储库中文件的版本。如果文件已添加标签，那么将原始标签并替换为新标签。

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

表 67. `setLabel` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中文件的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000010dcee0eaa28219"
<code>version</code>	必需	字符串	文件的特定版本	"0:2006-08-25 21:15:49.453"
<code>label</code>	必需	字符串	要应用于文件的标签	"Version 1"

表 68. `setLabel` 的返回值.

类型	描述
String	更新的文件的 URI

表 69. `setLabel` 的异常.

类型	描述
ResourceNotFoundException	源文件或版本不存在。
InsufficientParameterException	未指定必需参数。

## 示例

以下样本将标签 *Beta* 分配给文件 *MyReport.rptdesign* 的第二个版本。Resource 对象的 `getVersionMarker` 方法返回要添加标签的版本的标记。

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

## setMetadata 方法

将元数据属性应用于存储库中的文件和文件夹。

下表标识元数据属性及其是否可应用于文件和文件夹。

表 70. 存储库对象属性.

元数据属性	资源类型
作者	文件
描述	文件或文件夹
标题	文件或文件夹
到期日期	文件或文件夹
关键字	文件
主题	文件
定制元数据	文件或文件夹

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

表 71. `setMetadata` 的输入参数.

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	存储库中文件或文件夹的标准路径或对象 URI	"/Temp Folder/Temp.txt" 或 "0a58c3670016a7860000 010dcee0eaa28219"
<i>version</i>	可选。可指定版本或标签。	字符串	要下载的文件的具体版本	"0:2006-08-25 21:15:49.453"
<i>label</i>	可选。可指定版本或标签。	字符串	特定版本的标签	"Label 1"
<i>props</i>	必需	字典	包含字典中要设置的所有元数据，并使用元数据名称作为键。如"示例值"列中所示，以列表作为 <i>topic</i> 的值，并将字典用于 <i>customProperty</i> 。对于其余元数据，其获取字符串。	{ 'author': 'admin', 'title': 'newTitle', 'description': 'desc', 'topic': [a,b], 'customProperty': { 'language': 'hindi english', 'FreeForm': 'abcd' } }

表 72. *setMetadata* 的返回值.

类型	描述
String	针对其设置元数据的文件或文件夹的 URI

表 73. *setMetadata* 的异常.

类型	描述
InsufficientParameterException	未指定必需参数。
ResourceNotFoundException	源文件或文件夹不存在。

## 示例

以下样本将关键字 *Quarterly* 分配给文件 *MyReport.rptdesign* 的生产版本。

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

## uploadFile 方法

使用如果已存在则创建文件的新版本的选项，将文件从本地文件系统保存到存储库。

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

表 74. *uploadFile* 的输入参数.

字段	使用	类型	描述	示例值
<i>source</i>	必需	字符串	要上载的文件的标准路径（在本地文件系统中）	"C:\Temp\Temp.txt"
<i>target</i>	必需	字符串	要上载文件的存储库中目标文件夹的标准路径	"/Temp Folder"
<i>versionFlag</i>	可选	布尔值	如果指定的文件已存在，那么将创建文件的新版本	True 或 False

表 75. *uploadFile* 的返回值.

类型	描述
String	上载的文件的 URI

表 76. *uploadFile* 的异常.

类型	描述
ResourceNotFoundException	源文件或目标文件夹不存在。
ResourceAlreadyExistsException	目标文件夹中存在与源文件名称相同的文件或文件夹，并且未指定 <i>versionFlag</i> 参数。
InsufficientParameterException	未指定必需参数。

## 示例

此示例将文件 *MyReport.rptdesign* 上载到存储库中的 */Demo/Drafts* 文件夹。如果文件已存在，那么将使用 *versionFlag* 参数上载文件的新版本。

```

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

```

## Wrapper 类

PESImpl API 包含充当内容存储库方法调用的 Web service 返回的对象的包装器的类。这些 wrapper 类提供接口以显示方法返回的信息。

## Resource 类

Resource 类充当存储库对象 ResourceSpecifier.Resource 的简化包装器，提供对特定于对象的信息的访问。

除了与存储库对象相关联的标准元数据，该类还包含针对存储库中的对象定义的任何定制元数据信息。表 77 列出 Resource 类中可用的所有方法。

表 77. Resource 类的方法.

方法名称	描述
getAccessControlList	返回对象安全许可权的字典。其包含用户名作为键，并且仅将最高的许可权授予用户。例如： 如果用户 Joe 具资源 X 的删除许可权，那么表示 X 的资源对象的 getAccessControlList 将从 Web Service 调用返回 {'Joe': 'DELETE'}，而非所有三个许可权（读、写、删除）。
getOwner	返回对象所有者的名称作为字符串
getAuthor	返回对象作者的名称作为字符串
getContentSize	返回对象的大小
getCreatedBy	返回创建对象的用户的名称作为字符串
getCreationDate	返回对象的创建日期作为日期时间对象
getDescription	返回对象的描述作为列表
getDescriptionLanguage	返回对象的语言作为列表
getExpirationDate	返回对象的到期日期作为日期时间对象
isExpired	指示指定的对象是否已到期
getMimeType	返回对象的 MIME 类型作为字符串
getModificationDate	返回对象的上次修改日期作为日期时间对象
getObjectCreationDate	返回对象的对象创建日期作为日期时间对象
getObjectLastModifiedBy	返回上次修改对象的用户作为字符串
getObjectLastModifiedDate	返回对象的上次修改日期作为日期时间对象
getResourceID	返回对象的资源标识作为字符串
getResourcePath	返回指定对象的路径作为字符串
getTitle	返回对象的标题作为字符串
getTopicList	返回主题列表作为对象
getVersionMarker	返回对象的版本作为字符串
getVersionLabel	返回对象的标签作为字符串
getCustomMetadata	返回与对象相关联的任何定制属性作为字典

表 77. Resource 类的方法 (续).

方法名称	描述
getKeywordList	返回与对象相关联的关键词列表

## IdentificationSpecifier 类

该类充当存储库对象 IdentificationSpecifier 的简化包装器, 允许访问对象的特定于标识的数据。

表 78 列出 IdentificationSpecifier 类中可用的所有方法。

表 78. IdentificationSpecifier 类的方法.

方法名称	描述
getIdentifier	返回对象的标识值作为字符串
getVersionMarker	返回对象的版本作为字符串
getVersionLabel	返回应用于对象版本的标签作为字符串

## PageResult 类

PageResult 类充当搜索结果的容器。结果中的单个匹配项对应于 PageResult 对象中的一行。

例如, 返回四个资源的搜索将生成包含四个行的 PageResult 对象。表 79 列出 PageResult 类中可用的所有方法。

表 79. PageResult 类的方法.

方法名称	描述
getRows	返回 SearchRow 对象的列表。请参阅主题『SearchRow 类』以获取更多信息。

## SearchRow 类

SearchRow 类充当有关单个搜索结果的对象级别信息的容器。您可以使用此类的方法来访问有关对象的元数据。

表 80 列出 SearchRow 类中可用的所有方法。

表 80. SearchRow 类的方法.

方法名称	描述
getTitle	返回文件或文件夹的名称
getAuthor	返回文件或文件夹的作者
getMimeType	返回文件的 MIME 类型
getObjectLastModifiedBy	返回上次修改文件或文件夹的用户
getModified	返回上次修改文件或文件夹的日期和时间
getFolderPath	返回文件或文件夹的位置
getFolder	返回文件或文件夹的父文件夹的名称
getParentURI	返回父代的对象 URI
getTopic	返回与文件或文件夹相关联的主题
getChildRow	返回 SearchChildRow 对象的列表 (请参阅以下部分以获取更多信息)

要访问对象的版本级别的信息, 请使用 getChildRow 方法以返回与对象版本相对应的子行。

## SearchChildRow 类

SearchChildRow 类充当有关单个搜索结果的版本级别信息的容器。您可以使用此类的方法来访问有关对象版本的元数据。

表 81 列出 SearchChildRow 类中可用的所有方法。

表 81. SearchChildRow 类的方法.

方法名称	描述
getExpirationDate	返回文件或文件夹的到期日期
getKeyword	返回与文件或文件夹的版本相关联的关键词
getVersionLabel	返回文件或文件夹的版本标签
getDescription	返回文件或文件夹的描述
getLanguage	返回语言
getVersionCreationDate	返回创建文件或文件夹的日期和时间
getVersionMarker	返回文件或文件夹的版本标记
getUri	返回文件或文件夹的对象 URI

## 流程管理 API

流程管理脚本编制提供能力来处理作业。此区域包含以下功能：

- 执行作业
- 检索作业历史记录
- 检索作业详细信息

此部分概述用于处理存储库中存储的作业的 PESImpl 方法。每个方法都包含详细的语法信息、示例和期望的消息。

## 方法

以下部分列出 IBM SPSS Collaboration and Deployment Services 支持的所有流程管理脚本编制方法。

注：对于需要存储库中文件/文件夹路径的所有方法，可以使用路径或对象 URI。可以通过在 IBM SPSS Deployment Manager 中查看对象属性来获取对象 URI。

### cancelJob 方法

取消正在运行的作业。

`cancelJob(executionId)`

表 82. cancelJob 的输入参数.

字段	使用	类型	描述	示例值
<code>executionId</code>	必需	字符串	作业的执行标识	0a58c33d002ce90800 00010e0ccf7b01800e

表 83. cancelJob 的返回值.

类型	描述
Boolean	在作业取消时返回成功消息

## 示例

此示例终止报告作业的执行。

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

## deleteJobExecutions 方法

删除一个或多个作业执行。

`deleteJobExecutions(executionId)`

表 84. `deleteJobExecutions` 的输入参数.

字段	使用	类型	描述	示例值
<code>executionId</code>	必需	列表	作业执行标识的列表	[0a58c33d002ce9080000010e0ccf7b01800e, 0a59c33d002ce90800060010e0cdf7b01802r]

表 85. `deleteJobExecutions` 的返回值.

类型	描述
Boolean	True 或 False, 基于方法是否成功运行。

## 示例

此示例删除报告作业的执行。

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

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

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

## executeJob 方法

根据传递的参数并行或异步运行作业。在同步运行的情况下，方法直至作业完成才返回。在异步运行的情况下，方法在作业启动后返回。

`executeJob(source,notification,asynchronous)`

表 86. `executeJob` 的输入参数.

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	要上载的文件的标准路径（在本地文件系统中）	"C:\Temp\Temp.txt"
<code>notification</code>	可选	布尔值	指示是带通知运行作业还是不带通知。缺省值为 False。	True 或 False
<code>asynchronous</code>	可选	布尔值	指示是否异步运行作业。缺省值为 False。	True 或 False

表 87. `executeJob` 的返回值。

类型	描述
String	作业执行的唯一标识。此标识用于指示特定作业执行。

## 示例

此示例启动带通知的报告作业异步执行。

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

## getJobExecutionDetails 方法

针对特定作业列出运行详细信息，包括任何作业步骤和迭代。

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

表 88. `getJobExecutionDetails` 的输入参数。

字段	使用	类型	描述	示例值
<code>executionId</code>	必需	字符串	作业的执行标识	0a58c33d002ce9080000 010e0ccf7b01800e
<code>log</code>	可选	布尔值	指示是否内嵌显示作业日志	True 或 False
<code>target</code>	可选	字符串	存储日志的位置（在本地文件系统上）。仅与 <code>--log</code> 参数结合使用。	"c:\logs"

表 89. `getJobExecutionDetails` 的返回值。

类型	描述
<code>jobExecutionDetails</code>	有关作业执行的详细信息。请参阅主题第 54 页的『 <code>jobExecutionDetails</code> 类』以获取更多信息。

## 示例

此示例检索有关标识为 `execId` 的作业执行的作业步骤执行的信息，会将每个步骤的结果发送到控制台。

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

## getJobExecutionList 方法

针对作业的所有版本，列出特定作业的运行，包括任何当前正在运行的作业和已完成的作业。

```
getJobExecutionList(source)
```

表 90. `getJobExecutionList` 的输入参数。

字段	使用	类型	描述	示例值
<code>source</code>	必需	字符串	存储库中作业的标准路径。	"/testJob"



表 91. `getJobExecutionList` 的返回值。

类型	描述
PageResult	作业执行列表的容器。请参阅主题『PageResult 类』以获取更多信息。

## 示例

此示例检索报告作业的执行，会将有关每个执行的信息发送到控制台。

```
from pes.api.PESImpl import PESImpl
pesImpl = PESImpl("admin", "spss", "localhost", "8080")
executions = pesImpl.getJobExecutionList(source="/Demo/Reports")
execRows = executions.getRows()
if execRows:
    for exrow in execRows:
        print "Job Path: ", exrow.getPath()
        print "Object ID: ", exrow.getObjId()
        print "Event ID: ", exrow.getEventObjId()
        print "Version ", exrow.getVersionMarker()
        print "Started: ", exrow.getEventStartDateTime()
        print "Ended: ", exrow.getEventEndDateTime()
```

## Wrapper 类

PESImpl API 包含充当流程管理方法调用的 Web service 返回的对象的包装器的类。这些 wrapper 类提供接口以显示方法返回的信息。

## PageResult 类

PageResult 类充当作业执行结果的容器，允许检索特定于作业执行的数据。

单个作业执行对应于 PageResult 对象中的一行。例如，执行四次的作业对应于包含四行的 PageResult 对象。表 92 列出 PageResult 类中可用的所有方法。

表 92. PageResult 类的方法。

方法名称	描述
getRows	返回 Row 对象的泪飘，每个对应于一个作业执行。请参阅主题『Row 类』以获取更多信息。

## Row 类

Row 类充当有关作业执行的作业级别信息的容器。您可以使用此类的方法来访问有关作业执行的元数据。

表 93 列出 Row 类中可用的所有方法。

表 93. Row 类的方法。

方法名称	描述
getObjId	返回作业的执行标识
getPath	返回作业的路径
getVersionMarker	返回运行的作业的版本标记
getVersionLabel	返回运行的作业的版本标签
getEventObjId	返回运行的作业的事件标识
getEventState	返回正在运行的作业的状态
getEventCompletionCode	返回作业的完成代码

表 93. Row 类的方法 (续).

方法名称	描述
getEventStartDateTime	返回作业的开始日期和时间
getEventEndDateTime	返回作业的结束日期和时间
getQueuedDateTime	返回作业的排队日期和时间

## jobExecutionDetails 类

getJobExecutionDetails 方法返回该类。其存储作业的运行详细信息，并且包含提供有关作业中每个步骤的信息的 jobStepExecution 对象的列表。

表 94 列出 jobExecutionDetails 类中可用的所有方法。

表 94. jobExecutionDetails 类的方法.

方法名称	描述
getJobStepDetails	返回 jobStepExecutionDetails 对象的列表。请参阅主题『jobStepExecutionDetails 类』以获取更多信息。
getArtifactLocation	返回作业工件位置的列表
getCompletionCode	返回作业执行的完成代码
getEndTime	返回作业执行的结束日期和时间
getEventName	返回作业执行的事件名称
getEventUUID	返回作业执行的事件标识
getExecutionState	返回作业执行的运行状态
getExecutionSuccess	返回作业执行的成功或失败状态
getExecutionWarning	指示是否有任何警告
getLog	返回生成的日志（作为字符串）
getNotificationEnabled	指示是否启用电子邮件通知
getQueuedDateTime	返回作业执行的排队日期和时间
getStartDateTime	返回作业执行的开始日期和时间
getUserName	返回运行作业的用户名
getUUID	返回作业的执行标识

## jobStepExecutionDetails 类

该类存储作业步骤的运行详细信息，并存储 jobStepChildExecutionDetails 对象的列表。该类包含其授权所有方法调用的 ExecutionDetails 对象。

表 95 列出 jobStepExecutionDetails 类中可用的所有方法。

表 95. jobStepExecutionDetails 类的方法.

方法名称	描述
getJobStepChildExecutionList	返回 jobStepChildExecutionDetails 对象的列表。请参阅主题第 55 页的『jobStepChildExecutionDetails 类』以获取更多信息。
getArtifactLocation	返回作业步骤工件位置的列表
getCompletionCode	返回作业步骤的完成代码
getEndTime	返回作业步骤的结束日期和时间

表 95. *jobStepExecutionDetails* 类的方法 (续).

方法名称	描述
getEventName	返回作业步骤的事件名称
getEventUUID	返回作业步骤的事件标识
getExecutionState	返回作业步骤的运行状态
getExecutionSuccess	返回作业步骤的成功或失败状态
getExecutionWarning	指示是否有任何警告
getLog	返回生成的日志 (作为字符串)
getNotificationEnabled	指示是否启用电子邮件通知
getQueuedDateTime	返回作业步骤的排队日期和时间
getStartDateTime	返回作业步骤的开始日期和时间
getUserName	返回运行作业步骤的用户名
getUUID	返回作业步骤的执行标识

### jobStepChildExecutionDetails 类

*jobStepChildExecutionDetails* 类充当单个作业步骤的子执行的容器。例如, 迭代式报告作业步骤针对步骤的每个迭代生成子执行。您可以使用该类的方法访问有关子执行的元数据。

表 96 列出 *jobStepChildExecutionDetails* 类中可用的所有方法。

表 96. *jobStepChildExecutionDetails* 类的方法.

方法名称	描述
getArtifactLocation	返回子执行工件位置的列表
getCompletionCode	返回子执行的完成代码
getEndDateTime	返回子执行的结束日期和时间
getEventName	返回子执行的事件名称
getEventUUID	返回子执行的事件标识
getExecutionState	返回子执行的运行状态
getExecutionSuccess	返回子执行的成功或失败状态
getExecutionWarning	指示是否有任何警告
getLog	返回生成的日志 (作为字符串)
getNotificationEnabled	指示是否启用电子邮件通知
getQueuedDateTime	返回子执行的排队日期和时间
getStartDateTime	返回子执行的开始日期和时间
getUserName	返回子执行的用户名
getUUID	返回子执行的执行标识

---

## 示例脚本

以下目录找那个包含阐述如何 PESImpl 类的使用的示例脚本：

`<installation location>/samples`

这些脚本执行各种任务，包括以下任务：

- 从 IBM SPSS Collaboration and Deployment Services Repository 删除已到期的项
- 删除已到期的提交的工件
- 删除作业历史记录

您可以在 IBM SPSS Deployment Manager 中从"常规"作业步骤调用脚本来执行存储库维护任务。

---

## 声明

本信息是为在美国提供的产品和服务编写的。IBM 可能以其他语言提供本材料。然而，您可能需要拥有产品或产品版本的该语言副本才能进行访问。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您所在区域当前可获得的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并不意味着授予用户使用这些专利的任何许可。您可以用书面形式将许可查询寄往：

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

有关双字节 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

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

International Business Machines Corporation"按现状"提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些管辖区域在某些交易中不允许免除明示或暗含的保证，因此本条款可能不适用于您。

本信息可能包含技术方面不够准确的地方或印刷错误。本信息将定期更改；这些更改将编入本信息的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是本 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 使其能够在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及 (ii) 使其能够对已经交换的信息进行相互使用，请与下列地址联系：

*IBM Director of Licensing*  
*IBM Corporation*

North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

所引用的性能数据和客户示例仅作参考用途。实际的性能结果可能会因特定的配置和运营条件而异。

涉及非 IBM 产品的信息是从这些产品的供应商、已出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常业务经营中使用的数据和报告示例。为了尽可能全面地作举例说明，这些示例包含个人、公司、商标和产品的名称。所有这些名字都是虚构的，若与实际个人或业务企业相似，纯属巧合。

版权许可证：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例尚未在所有条件下经过全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。这些实例程序“按现状”提供，不附有任何种类的保证。对于因使用样本程序所引起的任何损害，IBM 概不负责。

---

## 隐私策略注意事项

IBM 软件产品（包括软件即服务（“软件产品”））可能使用 cookie 或其他技术来收集产品使用情况信息，从而帮助改善最终用户体验，定制与最终用户的交互或用于其他用途。在许多情况下，软件产品不会收集任何个人可标识信息。我们的一些软件产品可帮助收集个人可标识信息。如果此软件产品使用 cookie 来收集个人可标识信息，那么关于本产品使用 cookie 的具体信息，如下所述。

本软件产品不使用 cookie 或其他技术来收集个人可标识信息。

如果为此“软件产品服务”部署的配置允许客户通过 Cookie 和其他技术向最终用户收集个人标识信息，那么您应该寻求自己的法律建议，了解所有适用于此类数据收集的法律，包括所有对声明和同意的要求。

有关将各种技术（包括 Cookie）用于这些用途的更多信息，请参阅 IBM 隐私策略 (<http://www.ibm.com/privacy>) 和 IBM 网上隐私声明 (<http://www.ibm.com/privacy/details>) 中标题为“Cookie、Web Beacon 和其他技术”的部分以及“IBM Software Products and Software-as-a-Service Privacy Statement”(<http://www.ibm.com/software/info/product-privacy>)。

---

## 商标

IBM、IBM 徽标和 [ibm.com](http://www.ibm.com) 是 International Business Machines Corp.，在全球许多管辖区域的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。IBM 商标的最新列表可从 Web 上的“Copyright and trademark information”处获得，网址为：[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)。

Adobe、Adobe 徽标、PostScript 以及 PostScript 徽标是 Adobe Systems Incorporated 在美国和/或其他国家或地区的注册商标或商标。

Intel、Intel 徽标、Intel Inside、Intel Inside 徽标、Intel Centrino、Intel Centrino 徽标、Celeron、Intel Xeon、Intel SpeedStep、Itanium 和 Pentium 是 Intel Corporation 或其子公司在美国和其他国家或地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和@3B72其他国家或地区的注册商标。

Microsoft、Windows、Windows NT 以及 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和/或其他国家或地区的注册商标。

Java 和所有基于 Java 的商标和徽标是 Oracle 及/或其分支机构的商标和注册商标。

其他产品和服务名称可能是 IBM 或其他公司的商标。



Printed in China