

Feeder Steps

Allgemein

Erklärungen

Erklärungen zur Notation in den Codesequenzen.

Ausdruck	Bedeutung	Beispiel
[parameter]	Optionaler Parameter	[deleteInfected]
{ WERT1 wert2 wert3 }	Auswahl der gültigen Parameterwerte, default in Grossbuchstaben	{ TRUE false }

Hilfe-Output

Um die Parameter einer Operation zu sehen, kann die entsprechende Klasse ohne Parameter auf der Shell aus dem Verzeichnis %FEEDER_JAVA% aufgerufen werden, bspw. SIP-Extractor:

```
java ch.docuteam.feeder.qualityassurance.SIPExtractor
```

Ergibt folgenden Output:

```
ERROR 2014-01-02T10:54:25.140 (SIPExtractor) A wrong number of parameters
was passed to the command executor
INFO 2014-01-02T10:54:25.140 (SIPExtractor) Usage: java
ch.docuteam.documill.qualityassurance.SIPExtractor [path/to/]SIP
Parameters:
    [path/to/]SIP: name of the SIP; if not path is given, it will be
expected to be in the location defined by the docudarc.workbench.workdir ...
```

Admin

Admin: execute

Run a cmd command

```
Usage: cmd /C [SIP]
Parameters:
    SIP: cmd command
```

Admin: show feeder Version

Die Version gibt den Wert, der installierten Version der docuteam feeder Bibliothek und die Versions

Nummer von den abhängigen docuteam Bibliotheken, aus.

```
Usage: java ch.docuteam.feeder.admin.Version -v
docuteam feeder: 3.4.0 (19.04.2017)
docuteam darc: 2.18.7 (19.04.2017)
docuteam converter: 1.4.0 (10.10.2016)
docuteam tools: 1.11.8 (18.01.2017)
```

Ingest

Ingest: BARSIP Converter

Der BARSIPConverter konvertiert ein BARSIP SIP in ein SIP welches dem Matterhorn Profile entspricht.

```
Usage: java ch.docuteam.feeder.ingest.BARSIPConverter [path/to/]BAR-SIP
[targetFolder]
Parameters:
    [path/to/]BAR-SIP: name of the BAR-SIP folder or ZIP-file; if no path is
given, it will be expected to be in the location defined by the
'feeder.workbench.dropbox' property
    [targetFolder]: directory where to move the created SIP to; if omitted,
the SIP will be moved to the location defined by the
'feeder.workbench.workdir' property
```

Ingest: Check Workbench Space

Prüft, ob genügend Platz für die Verarbeitung des SIPs (d.h. für Arbeitskopien) vorhanden ist.

```
Usage: java ch.docuteam.feeder.ingest.CheckWorkbenchSpace [path/to/]SIP
[numberOfCopies]
Parameters:
    [path/to/]SIP: name of the SIP. If not path is given, it will be
expected to be in the location defined by the feeder.workbench.work property
    [numberOfCopies]: optional, number of copies to calculate with; defaults
to '3'
```

Ingest: cleanup working copies

Der Cleanup wird gebraucht um vorhandene SIP(s) aus dem Workfodler in der Workbench zu löschen. Dabei wird die Workbench welche in der Datei docuteamFeeder.properties.file definiert ist verwendet. Als Option, können auch SIP Versionen welche im Prepration Verzeichnis in der Workbench liegen und den gleichen Namen haben gelöscht werden.

```
Usage: java ch.docuteam.feeder.ingest.Cleanup [path/to/]SIP [prep]
Parameters:
```

```
[path/to/]SIP: name of the SIP. If not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property
```

```
[prep]: if 'true', SIPs of the same name in the preparation folder will be removed as well; defaults to 'false'
```

Ingest: create EAD file

CreateEADFile erstellt aus einzelnen Knotenpunkten von einem gegebenen SIP, EAD Datenblöcke und legt diese in ein fertig erstellte Verzeichnis ab.

```
Usage: java ch.docuteam.feeder.ingest.CreateEADFile [path/to/]SIP
```

Parameters:

```
[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property
```

Ingest: extent calculator

Setzt die Anzahl Dateien in das Metadatenfeld „Umfang“ und die Einheit auf den Default „Datei(en)“.

```
Usage: java ch.docuteam.feeder.ingest.ExtentCalculator [path/to/]SIP
```

Parameters:

```
[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.work property
```

Ingest: file migration

The SIPFileMigrator compares the files of a SIP with the settings of a configuration file (migration-config.xml) and converts the files according to the definitions in that file.

```
Usage: java ch.docuteam.feeder.ingest.SIPFileMigrator [path/to/]SIP keepOriginals [path/to/migration-config.xml]
```

Parameters:

```
[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.work property keepOriginals: { true | false }, indicating whether to keep the original files after the migration process
```

```
[path/to/migration-config.xml]: optional path to a specific migration configuration file (defaults to ./config/migration-config.xml)
```

Ingest: remove SIP

Die Funktion SIP RemoveFromInbox löscht ein vorhandenes SIP aus der Inbox und legt dieses dann in den gegebenen Ordner ab oder löscht es sofern kein Zielordner gegeben ist.

```
Usage: java ch.docuteam.feeder.ingest.SIPRemoveFromInbox [path/to/]SIP
[targetFolder]
```

Parameters:

[path/to/]SIP: path of the SIP; if only the name is given, it will be expected to be in the location defined by the feeder.workbench.inbox property

[targetFolder]: directory where to move the SIP to; if omitted, the SIP will be deleted

Ingest: Remove SIP from Inbox

Der Schritt SIPRemoveFromInbox operation löscht ein vorhandenes SIP aus der Inbox in einen vorgegebenen Ordner oder löscht es, sofern kein Zielordner angegeben ist.

```
Usage: java ch.docuteam.feeder.ingest.SIPRemoveFromInbox [path/to/]SIP
[targetFolder]
```

Parameters:

[path/to/]SIP: path of the SIP; if only the name is given, it will be expected to be in the location defined by the feeder.workbench.inbox property

[targetFolder]: directory where to move the SIP to; if omitted, the SIP will be deleted

Ingest: transform EAD

<http://www.saxonica.com/html/documentation/javadoc/net/sf/saxon/Transform.html>

Beispiel: EAD transformation in AIS spezifisches Format.

```
java net.sf.saxon.Transform -s://path/to/workbench/2_work/${SIP}/EAD.xml -
xsl:C:/docuteam/apps/feeder_java/resources/xslt/EAD2AIS.xslt -
o://path/to/workbench/7_AIS-EADFiles/${SIP}.xml
```

Quality Assurance

Quality Assurance: extract SIP into workfolder

Extrahiert ein gezipptes SIP in den workfolder der Workbench. Ein optionales zweites Argument kann verwendet werden um einen anderen Zielordner anzugeben.

```
Usage: java ch.docuteam.feeder.qualityassurance.SIPExtractor [path/to/]SIP
[targetdir]
```

Parameters:

[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir

property

[targetdir]: target directory; absolute path of the directory where to unzip the SIP to. Optional, default to the workdir defined in the docuteamFeeder.properties

Quality Assurance: fixity check (md5)

Überprüft die Dateien auf Konformität mit den hinterlegten Checksummen in der METS Datei. Die Resultate der Prüfung werden in Form von PREMIS Events als inline xml Code in das METS File geschrieben.

Usage: java ch.docuteam.feeder.qualityassurance.SIPFixityCheck [path/to/]SIP
Parameters:
[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property

Quality Assurance: file path length check

Prüft, ob die Länge von absoluten Pfaden eines SIPs über einem anzugebenden Wert liegen.

Usage: java ch.docuteam.feeder.qualityassurance.FilePathLengthCheck /absolute/path/to/folder maxAllowedFilePathLength
Parameters:
/absolute/path/to/folder: absolute path of the folder that should be checked
maxAllowedFilePathLength: the max allowed number of characters of the canonical file path

Quality Assurance: sip path length check

Prüft die Dateipfadlängen innerhalb eines SIPs gegen einen anzugebenden Grenzwert.

Usage: java ch.docuteam.feeder.qualityassurance.SIPPathLengthCheck [path/to/]SIP maxAllowedFilePathLength
Parameters:
[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.work property
maxAllowedFilePathLength: the max allowed number of characters of the canonical file path

Quality Assurance: get PID

Verbindet das Fedora Repository und holt eine einzelne PID um das SIP zu identifizieren. In der Sequenz, wird das entsprechende PID als Haupt-Einfalls-Tor in das Repository für die Submission

verwendet. Der Wert wird in dem <mets:OBJID> Element eingelagert.

```
Usage: java ch.docuteam.feeder.qualityassurance.SIPConfirmation
[path/to/]SIP [PIDNamespace[:###]]
Parameters:
    [path/to/]SIP: name of the SIP. If not path is given, it will be
expected to be in the location defined by the feeder.workbench.workdir
property
    [PID namespace[:###]]: namespace for new PID or complete PID to use for
the object; if omitted, the standard namespace from the submission agreement
will be used; if the submission agreement cannot be found, the default
namespace of the Fedora repository will be used.
```

Quality Assurance: convert to safe filenames

Benennt Dateien mit Spezialzeichen um. Sichere Dateinamen beinhalten nur Zeichen aus A-Z, a-z, 0-9, und „_“-.

```
Usage: java ch.docuteam.feeder.qualityassurance.SIPConvertToSafeFileNames
[path/to/]SIP
Parameters:
    [path/to/]SIP: name of the SIP; if not path is given, it will be
expected to be in the location defined by the feeder.workbench.work property
```

Quality Assurance: delete backup files

Löscht Dateien aus dem SIP die einem bestimmten Namensmuster entsprechen.

```
Usage: java ch.docuteam.feeder.qualityassurance.SIPDeleteBackupFiles
[path/to/]SIP [filenamePattern filenamePattern ...]
Parameters:
    [path/to/]SIP: name of the SIP; if not path is given, it will be
expected to be in the location defined by the feeder.workbench.work property
    [filenamePattern filenamePattern ...]: a list of filename patterns (NOT
case-sensitive, '*' is wildcard, but is only allowed at the beginning or end
of the pattern). Files matching any one of this patterns will be deleted
```

Quality Assurance: SIPSubmissionAgreementCheck

Prüft ob die Dateiformate mit den Vorgaben aus dem Submission Agreement übereinstimmen. Es gibt zwei Modis: im ersten Modus (removeBadFiles = false), wird jede Datei welche über keine Übereinstimmung mit dem Submission Agreement hat aufgelistet (verwendet werden hierzu die WARN log Einträge) und zudem wir dein Fehlercode ausgegeben.

Beim zweiten Modus (removeBadFiles = true), wird jede Datei welche über keine Übereinstimmung mit dem submission agreement aufweist, aus dem SIP gelöscht. Das modifizierte METS.xml wird hier

gespeichert (das originale SIP bleibt unverändert als Backup bestehen).

```
Usage: JAVA ch.docuteam.feeder.qualityassurance.SIPSubmissionAgreementCheck
[path/to/]SIP [removeBadFiles]
```

Parameters:

[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property

[removeBadFiles]: optional, { true | false }; indicating whether to automatically remove files that are not valid according to the submission agreement

Quality Assurance: SIP virus check

Jede im SIP vorhandene Datei wird auf Viren überprüft. Für die Virenprüfung wird der Virens Scanner von ClamAV (www.clamav.net) verwendet.

Voraussetzung für diese Prüfung ist ein gestartet ClamAV Dienst. Abhängig vom zweiten Argument werden infizierte Dateien verworfen oder automatisch gelöscht.

```
Usage: java ch.docuteam.feeder.qualityassurance.SIPVirusCheck [path/to/]SIP
deleteInfected Parameters:
```

[path/to/]SIP: name of the SIP; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property

deleteInfected: if 'true', the operation automatically removes infected files from the SIP

Storage

Storage: check checksums

Der ChecksumChecker vergleicht die Objekte aus dem ORIGINAL Datastream in Fedora mit der generierten Checksumme. Die FEEDER_JAVA System Variable wird dabei zur Lokalisierung der Konfigurations Dateien verwendet.

```
Usage: java ch.docuteam.feeder.storage.ChecksumChecker -e
"[mailto:]<recipient>" [-s subjectOk|subjectError|subjectNoConnection] [-n
namespace..]
```

Parameters:

-e (or --email) <recipient>: URL of type mailto:recipient@example.com. If protocol is omitted it is automatically prepended to the email address

Optional Parameters:

-s (or --subject) subjectOk|subjectError|subjectNoConnection: mail subjects to use for the given outcomes, separated by '|'. Exactly 3 subjects must be given, but may be empty to use default. First subject is used, if everything is ok; second, if bad files are found; third, if no connection could be established to fedora. Use %ld to display the number of failed checks and %2s to display the URL of the targeted fedora.

```
-n (or --namespace) [namespace] [namespace] ...: Fedora namespaces
separated by space; if no namespace is given, all namespaces are checked.
```

Storage: create Fedora objects

Der FOXMLCreator konvertiert ein zugewiesenes METS Packet in eine separates FXOML (Fedora Object) Datei. Verwendet werden die aktuellen Verzeichnisse oder - falls nicht verfügbar - die FEEDER_JAVA Systemvariablen um die Konfigurations-Dateien und die in der Workbench definierten docuteamFeeder.properties Datei zu lokalisieren. Der Code benutzt das Verzeichnis des Ingest Tools welches auf der Fedora website verfügbar ist. Änderungen werden zur Unterstützung gemacht um die Unterschiede zwischen Root-Ordner und Ordner durchgehende Umsetzungs-Regeln (crules.xml) und um bereits während dem Ingest Prozess erhaltene PID zu handhaben.

```
Usage: java ch.docuteam.feeder.storage.FOXMLCreator [path/to/]SIP
Parameters:
    [path/to/]SIP: name of the SIP. If not path is given, it will be
expected to be in the location defined by the feeder.workbench.workdir
property
```

Storage: deliver DIP

Der DIPDeliverer erhält den/die Datastreams(s) für ein bereitgestelltes fedora PID oder eine Datei im PUID Format, diese wird wiederum als DIP verpackt und wird in dem zugewiesenen Verzeichnis gelagert.

```
Usage: java ch.docuteam.feeder.storage.DIPDeliverer['pid'|'puid'] [PID|PUID]
[targetLocation]
Parameters:
    ['pid' | 'puid'] (if 'pid' then provide a fedora PID, if 'puid' then
provide a pronom PUID)
    [PID|PUID] (PID = fedora persistent unique identifier, PUID = pronom
persistent unique identifier)
Optional Parameters:
    [targetLocation]: Location where to save DIP.
```

Storage: transfer Fedora objects to repository

Der FOXMLIngester überträgt die gegebene Liste von FOXML (Fedora Objekt) Dateien auf ein Fedora repository Storage. Die FEEDER_JAVA System Variable wird verwendet um Konfigurations Dateien und die definierte Workbench zu lokalisieren welche in der Datei docuteamFeeder.properties abgelegt ist.

```
Usage: java ch.docuteam.feeder.storage.FOXMLIngester [path/to/]SIP
[keepFOXML]
Parameters:
    [path/to/]SIP: name of the SIP. If no path is given, it will be expected
```



```
to be in the location defined by the feeder.workbench.finished property
keepFOXML: One of { true | false }, indicating whether to keep the FOXML
files after a successful ingest; defaults to 'true'
```

Storage: update Fedora object

Diese Operation lädt neue Versionen von einem Objekt auf den Fedora Server.

```
Usage: java ch.docuteam.feeder.storage.FedoraObjectUpdater [path/to/]SIP
Parameters:
    [path/to/]SIP: name of the SIP. If not path is given, it will be
expected to be in the location defined by the feeder.workbench.workdir
property
```

Storage: validate METS

Der METS Validator validiert die mets xml Dateien mit dem verbundenen Schema Definitionen und platziert die Definition des Namenraumes vom Root Element zu den jeweiligen Elementen. Dies ist eine notwendige Vorbereitung für das mets xml wenn dieses in unterschiedliche Teile aufgesplittet werden muss wenn bspw. unterschiedliche foxml Dateien aus einem SIP erstellt werden sollen (FOXMLCreator).

```
Usage: java ch.docuteam.feeder.storage.METSValidator [path/to/]SIP [withEAD]
Parameters:
    [path/to/]SIP: path of the SIP; if not path is given, it will be
expected to be in the location defined by the feeder.workbench.workdir
property
    [withEAD]: whether to include EAD as descriptive metadata and create a
datastream in the fedora objects; defaults to false
```

Storage: PID assigner

Weisst allen Knoten eines SIPs eine PID ab Fedora zu.

```
Usage: java ch.docuteam.feeder.storage.PIDAssigner [path/to/]SIP
Parameters:
    [path/to/]SIP: name of the SIP. If no path is given, it will be expected
to be in the location defined by the feeder.workbench.work property
```

Storage: PIDListPublisher

The PIDListPublisher saves/sends the file 'PIDs.txt' which results from the FOXMLIngester class to a given URL.

```
Usage: java ch.docuteam.feeder.storage.PIDListPublisher [path/to/]SIP
```

receiverURL

Parameters:

[path/to/]SIP: path of the SIP; if no path is given, it will be expected to be in the location defined by the `feeder.workbench.finished` property
receiverURL: An URL in the style of { file: | mailto: }, indicating whether to put/send the list with PIDs.

Storage: RenameSIPasAIPforlaas

The operation `RenameSIPasAIPforlaas` renames an SIP using the pid of the root element of the mets file as prefix. Needs 'PID' as accessorname in levels.xml.

Usage: `java ch.docuteam.feeder.storage.RenameSIPasAIPforIaaS [path/to/]SIP [targetFolder]`

Parameters:

[path/to/]SIP: path of the SIP; if only the name is given, it will be expected to be in the location defined by the `feeder.workbench.dropbox` property

[targetFolder]: directory where to put the AIP to; if omitted, the AIP will be copied to standard output directory '4_output'

Storage: UpdateExcelWithPID

This class will write PIDs from an SIP's nodes into excel sheet(s). The excel sheet(s) must have a column with a label of either 'identifier' or 'id' in the first row. PIDs will be written into the column with the header string 'PID' or – if such a column is not available – into the next free column.

Usage: `java ch.docuteam.feeder.storage.UpdateExcelWithPID [path/to/]SIP path/to/folder/with/excel`

Parameters:

[path/to/]SIP: path to the SIP; if relative path is given, try to find it in the workbench's working directory

path/to/folder/with/excel: path to the excel files to be updated

Storage: WebgateDigitalObjectUpdater

The `WebgateDigitalObjectUpdater` checks one or several curator databases for units with digital objects and whether they should be (de-)published on a given Fedora instance according to `docuteamFeeder.properties`. It will use the `FEEDER_JAVA` system variable to locate configuration files.

Usage: `java ch.docuteam.feeder.storage.WebgateDigitalObjectUpdater {*|db1,db2,..} [targetDirectory]`

Parameters:

{*|db1,db2}: Asterisk for all databases of the given database server or comma separated list of database names (wildcards '*' are allowed)

Optional Parameters:

[targetDirectory]: The directory, where the DIPs are stored. If omitted, the given path in docuteamFeeder.properties is used

Submission

Submission: AgreementsOverviewGenerator

Creates a simple overview of submission agreements located in a given folder. This is done by XSL transformations, for which the class looks for any submission agreement files in the given directory and lists them in a simple xml structure:

```
<safilelist>
  <sa_1 />
  <sa_2 />
  ...
  <sa_x />
</safilelist>
```

Usage: java ch.docuteam.feeder.submission.AgreementsOverviewGenerator
agreements_directory type output_directory

Parameters:

agreements_directory: location where the collection of submission agreements can be found

type: one of { Hierarchy | Flat | CSV }, defining the structure of the resulting overview file

output_directory: target location for the created overview file, defaults to the directory, where the agreements are located (args[0])

Submission: CheckFolder

Check if sip size and size of each file in folder and file paths within the SIP exceeds the maximal allowed provided value.

Any file or folder exceeding the maximal allowed size will be logged.

Usage: java ch.docuteam.feeder.submission.CheckFolder [/path/to/]folder
maxTotalSize maxSingleFileSize maxFilePathLength

Parameters:

[/path/to/]folder: path of the folder to check; if not path is given, it will be expected to be in the location defined by the feeder.workbench.workdir property

maxTotalSize: the max allowed size the folder may have

maxSingleFileSize: the max allowed size any of the files contained by the sip

maxFilePathLength: the max allowed length of file paths within the folder

Submission: CreateSIPFromExcel

Creates a SIP according to the Matterhorn METS profile, getting structure and descriptive metadata from an Excel sheet.

Preconditions are

- The first excel sheet in an excel workbook is asumed to be the sheet to be read in
- This sheet must have a column named path that contains all files and folders to be packed into the sip
- The paths can be relativ or absolute, it is not allowed to mix relativ and absolute paths, either these are relative or absolute
- A column with name levelOfDescription is expected. Only levels that exist in levels.xml are allowed
- Only metadata elements defined in levels.xml for the respective level are allowed, not defined metadata elements are reported as as a warn message in logging

```
Usage: java ch.docuteam.feeder.submission.CreateSIPFromExcel  
[path/to/]Excelfile saID dssID [path/to/target/directory]
```

Parameters:

```
[path/to/]Excelfile: name or path without file extension to the excel  
file; defaults to workbench/0_preparation if path is omitted  
saID: string that is used to reference a submission agreement  
dssID: string that is used to reference a data submission session within  
the submission agreement  
[path/to/target/directory]: path to the directory, where the SIP should  
be placed; optional, defaults to 'workbench/1_dropbox'
```

Submission: CreateSIPsFromFileOrFolder

The CreateSIPsFromFileOrFolder operation will create SIPs from a given file or folder. If the source is a folder, a parameter will define whether a single SIP or separate SIPs for each child should be created.

```
Usage: java ch.docuteam.feeder.submission.CreateSIPsFromFileOrFolder source  
recursive saID dssID author zipped [outputDir]
```

Parameters:

```
source: file or folder for which an SIP should be generated  
split: if 'true', a separate SIP will be created for each  
file/folder within the source (assuming the source is a folder)  
saID: value to use for referencing a submission agreement in the SIP  
dssID: value to use for referencing a data submission session of the  
respective submission agreement  
author: value to use as the creator for the SIP  
zipped: if 'true', create zipped SIPs  
[outputDir]: optional location where to put the SIPs; if omitted the  
property 'feeder.workbench.workdir' defined in the docuteamFeeder.properties  
will be used
```

Submission: SubmitSIPsFromFolder

The SubmitSIPsFromFolder will use the given arguments for selecting SIPs in a folder and submitting them to a number of workflows using the feeder REST-interface.

```
Usage: java ch.docuteam.feeder.submission.SubmitSIPsFromFolder inbox
errorbox filter feeder_url workflows user password useAbsolutePaths
checkEmptyQueue [maxNumberSIPs]
Parameters:
    inbox: path to the folder containing the SIPs
    errorbox: path to the folder where to put unsuccessful SIPs
    filter: regex filter string for the SIPs within the dropbox; put the
regex expression into quotation marks!
    feeder_url: URL pointing to the feeder main page, f.ex.
http://localhost/feeder
    workflows: comma separated list of workflows to execute on each SIP
    user: username for feeder
    password: password for feeder
    useAbsolutePaths: true/false, indicating whether to submit SIPs by
absolute paths or just their filenames
    checkEmptyQueue: true/false, indicating whether to check if the queue is
empty before submitting new SIPs
    [maxNumberSIPs] (optional): maximum number of SIPs to send to feeder; if
omitted, all SIPs matching the filter string will be submitted
```

Submission: WebjaxeAgreementCollector

The WebjaxeAgreementCollector will look for the any submission agreements created within the webjaxe editor and copy them to a given directory.

```
Usage: java ch.docuteam.feeder.submission.WebjaxeAgreementCollector
target_directory [webjaxe_home]
Parameters:
    target_directory: location where to store the submission agreements xml
files
    [webjaxe_home]: optional location of the webjaxe installation directory,
usually in the webserver's htdocs directory. If omitted, the environment
variable $WEBJAXE_HOME will be used.
```

Util

Util: MailSender

This MailSender operation sends an email to the given recipient with optional attachments.

```
Usage: java ch.docuteam.feeder.util.MailSender receiver subject text
```

```
[attachment1 [attachment2 [...] ] ]
```

Parameters:

receiver: the receiver's e-mail address(es), comma separated if several

subject: the mail subject

text: the message text

attachments: filepaths to attachments; if the first attachment is 'zip', all attachments will be zipped into a single attachment

From:

<https://wiki.docuteam.ch/> - **docuteam wiki**

Permanent link:

https://wiki.docuteam.ch/doku.php?id=docuteam:feeder-steps_340&rev=1508509949

Last update: **2019/01/07 11:16**

