



**JOB**SCHEDULER

# J A D E

## JS Advanced Data Exchange

Parameter Reference  
July 2014

## Contact Information

Software- und Organisations-Service GmbH

Giesebrechtstr. 15  
D-10629 Berlin  
Germany

Telephone +49 (0)30 86 47 90-0  
Telefax +49 (0)30 8 61 33 35  
Mail [info@sos-berlin.com](mailto:info@sos-berlin.com)  
Web <http://www.sos-berlin.com>

Last Updated: 07/18/2014 12:03 PM

This documentation is based on JobScheduler Version 1.7.4169.

**Copyright © 2005-2014 SOS GmbH Berlin.**

All rights reserved. All trademarks or registered trademarks are the property of their respective holders. All information and materials in this book are provided "as is" and without warranty of any kind. All information in this document is subject to change without further notice.

This product includes software developed by the Apache Software Foundation (<http://apache.org/>)

We would appreciate any feedback you have, or suggestions for changes and improvements; please forward your comments to [info@sos-berlin.com](mailto:info@sos-berlin.com).

# Table of Contents

- 1 Introduction ..... 8**
- 2 JADE-Client Parameters ..... 9**
  - 2.1 Transfer Profile Parameters ..... 9
    - 2.1.1 settings - Name of INI-File which contains the profiles to execute ..... 9
    - 2.1.2 profile - The Name of a Profile-Section to be executed ..... 10
    - 2.1.3 operation - Operation to be executed ..... 10
    - 2.1.4 verbose - The granularity of (Debug-)Messages ..... 10
    - 2.1.5 protocol - Type of requested Datatransfer ..... 10
  - 2.2 Connection Parameters ..... 11
    - 2.2.1 host - Host-Name ..... 11
    - 2.2.2 port - Port-Number to be used for Data-Transfer ..... 11
    - 2.2.3 protocol - Type of requested Datatransfer ..... 11
    - 2.2.4 proxy parameters ..... 11
      - 2.2.4.1 alternative\_host ..... 11
      - 2.2.4.2 alternative\_port ..... 12
  - 2.3 Authentication Parameters ..... 12
    - 2.3.1 user - UserID of user in charge ..... 12
    - 2.3.2 password - Password for UserID ..... 12
    - 2.3.3 Alternative Authentication, ..... 12
      - 2.3.3.1 alternative\_user ..... 12
      - 2.3.3.2 alternative\_password ..... 12
    - 2.3.4 SSH only Parameters ..... 13
      - 2.3.4.1 auth\_file ..... 13
      - 2.3.4.2 auth\_method ..... 13
      - 2.3.4.3 strict\_hostkey\_checking - SSH HostKey checking ..... 13
      - 2.3.4.4 SSH proxy parameter ..... 13
  - 2.4 Selecting Objects to Transfer ..... 14
    - 2.4.1 file\_path ..... 14
    - 2.4.2 file\_spec - File filtering using a regular expression ..... 15
    - 2.4.3 file\_list\_name - Name of a file which contains a filelist ..... 15
  - 2.5 Transactional Transfer ..... 15
    - 2.5.1 transactional - Transaction controlled Transfer ..... 15
    - 2.5.2 atomic\_suffix - Suffix for the target transfer file name ..... 16
    - 2.5.3 atomic\_prefix - Prefix for the target transfer file name ..... 16
  - 2.6 Atomic Transfer ..... 16
    - 2.6.1 atomic\_suffix - Suffix for the target transfer file name ..... 17
    - 2.6.2 atomic\_prefix - Prefix for the target transfer file name ..... 17
  - 2.7 Operations on Objects ..... 17
    - 2.7.1 Concurrent Transfer Parameters ..... 17
      - 2.7.1.1 concurrent\_transfer - Activate parallel transfers ..... 17
      - 2.7.1.2 max\_concurrent\_transfers - Max number of parallel transfers ..... 18
    - 2.7.2 Compressing Files ..... 18
      - 2.7.2.1 compress\_files ..... 18
      - 2.7.2.2 compressed\_file\_extension - Additional file-name extension for compressed files ..... 18
      - 2.7.2.3 decompress\_after\_transfer - Unpack Files after Transfer ..... 18
    - 2.7.3 Cumulate File Parameters ..... 18
      - 2.7.3.1 cumulate\_files - cumulate (all) files into a single file ..... 19

- 2.7.3.2 cumulative\_filename - cumulate (all) files into a single file ..... 19
- 2.7.3.3 cumulative\_file\_separator - cumulate (all) files into a single file ..... 19
- 2.7.3.4 cumulative\_file\_delete - cumulate (all) files into a single file ..... 20
- 2.7.4 Changing File Names ..... 20
  - 2.7.4.1 replacing - Regular expression for renaming Files ..... 20
  - 2.7.4.2 replacement - String for renaming Filenames ..... 22
  - 2.7.4.3 Examples: Cumulate Files ..... 22
- 2.7.5 Encoding ..... 24
  - 2.7.5.1 file\_name\_encoding - Set the encoding-type of a file name ..... 24
  - 2.7.5.2 control\_encoding - Specify the encoding-type used by the server ..... 24
- 2.8 Data Source only Parameters ..... 25
  - 2.8.1 force\_files - Raise Error, if no files found for transfer ..... 25
  - 2.8.2 recursive - Include subfolders in operation ..... 25
  - 2.8.3 local\_dir - local\_dir ..... 25
  - 2.8.4 remove\_files ..... 26
  - 2.8.5 zero\_byte\_transfer - Transfer of "zero byte" size files ..... 26
  - 2.8.6 Changing File Names ..... 26
    - 2.8.6.1 replacing - Regular expression for renaming Files ..... 26
    - 2.8.6.2 replacement - String for renaming Filenames ..... 28
    - 2.8.6.3 Examples: Cumulate Files ..... 29
  - 2.8.7 Check the completeness of a file (steady state) ..... 30
    - 2.8.7.1 check\_steady\_state\_of\_files - Check the completeness of a file (steady state) ..... 30
    - 2.8.7.2 steady\_state\_count - Maximum Number of Checkpoints ..... 31
    - 2.8.7.3 check\_steady\_state\_interval - temporal distance between checkpoints ..... 31
- 2.9 Data Target Parameters ..... 31
  - 2.9.1 transactional - Transaction controlled Transfer ..... 31
  - 2.9.2 append\_files - Append content of transferred files to existing files ..... 32
  - 2.9.3 overwrite\_files - Overwrite existing files ..... 32
  - 2.9.4 make\_dirs - Create folder ..... 32
  - 2.9.5 atomic\_prefix - Prefix for the target transfer file name ..... 33
  - 2.9.6 atomic\_suffix - Suffix for the target transfer file name ..... 33
  - 2.9.7 remote\_dir - Name of the Target Folder ..... 33
  - 2.9.8 alternative\_remote\_dir ..... 33
  - 2.9.9 strict\_hostkey\_checking - SSH HostKey checking ..... 33
  - 2.9.10 Changing File Names ..... 34
    - 2.9.10.1 replacing - Regular expression for renaming Files ..... 34
    - 2.9.10.2 replacement - String for renaming Filenames ..... 35
    - 2.9.10.3 Examples: Cumulate Files ..... 36
  - 2.9.11 Cumulate File Parameters ..... 38
    - 2.9.11.1 cumulate\_files - cumulate (all) files into a single file ..... 38
    - 2.9.11.2 cumulative\_filename - cumulate (all) files into a single file ..... 38
    - 2.9.11.3 cumulative\_file\_separator - cumulate (all) files into a single file ..... 38
    - 2.9.11.4 cumulative\_file\_delete - cumulate (all) files into a single file ..... 39
    - 2.9.11.5 Examples: Cumulate Files ..... 39
- 2.10 Parameters for Pre- and Post-Processing ..... 39
  - 2.10.1 Commands for the transfer ..... 40
    - 2.10.1.1 pre\_transfer\_commmands - pre\_transfer\_commmands ..... 40
    - 2.10.1.2 post\_transfer\_commmands - commands, which has to be executed after the transfer ended ..... 40
    - 2.10.1.3 Cumulate File Parameters ..... 40
    - 2.10.1.4 Compressing Files ..... 42
  - 2.10.2 Compressing Files ..... 43
    - 2.10.2.1 Pre\_Command - Command to be executed before transfer ..... 43

- 2.10.2.2 Post\_Command - Command to be executed after transfer ..... 43
- 2.11 FTP only Parameters ..... 43
  - 2.11.1 account - Additional accounting information ..... 43
  - 2.11.2 passive\_mode - passive\_mode ..... 43
  - 2.11.3 transfer\_mode - Type of Character-Encoding ..... 43
  - 2.11.4 alternative\_account - Additional accounting information ..... 44
  - 2.11.5 alternative\_passive\_mode ..... 44
  - 2.11.6 alternative\_transfer\_mode ..... 44
  - 2.11.7 check\_server\_features - Get a List of Server features ..... 44
  - 2.11.8 protocol\_command\_listener - Activate the ftp-client logging ..... 44
- 2.12 SSH only Parameters ..... 45
  - 2.12.1 auth\_file ..... 45
  - 2.12.2 auth\_method ..... 45
  - 2.12.3 strict\_hostkey\_checking - SSH HostKey checking ..... 45
  - 2.12.4 SSH proxy parameter ..... 45
    - 2.12.4.1 ssh\_proxy\_host ..... 45
    - 2.12.4.2 ssh\_proxy\_port - Port number of proxy ..... 46
    - 2.12.4.3 ssh\_proxy\_user - User-ID for the Proxy-Server ..... 46
    - 2.12.4.4 ssh\_proxy\_password - Password of Proxy-Server ..... 46
- 2.13 Params to control the Transfer History ..... 46
  - 2.13.1 Background Service Parameters ..... 46
    - 2.13.1.1 scheduler\_host ..... 46
    - 2.13.1.2 scheduler\_job\_chain - The name of a job chain ..... 46
    - 2.13.1.3 scheduler\_port ..... 47
    - 2.13.1.4 send\_transfer\_history - send transfer history to the background service ..... 47
    - 2.13.1.5 scheduler\_transfer\_method - how to communicate with the BackgroundService ..... 47
  - 2.13.2 Params to create a history file ..... 47
    - 2.13.2.1 history - The name of the transfer history file ..... 47
    - 2.13.2.2 history\_entries - additional entries for the transfer history ..... 49
    - 2.13.2.3 history\_repeat ..... 49
    - 2.13.2.4 history\_repeat\_interval ..... 49
    - 2.13.2.5 mandator ..... 49
- 2.14 Polling Parameters ..... 49
  - 2.14.1 poll\_interval - wait time interval between two poll-actions ..... 50
  - 2.14.2 poll\_minfiles - Minimum Number of Files expected ..... 50
  - 2.14.3 poll\_timeout - Max time to poll for files ..... 50
  - 2.14.4 poll\_keep\_connection - Keep connection while polling ..... 50
  - 2.14.5 waiting\_for\_late\_comers - Warten auf Nachzügler ..... 50
  - 2.14.6 poll\_error\_state - Name of next node in job-chain if polling failed ..... 50
- 2.15 Virtual File System Parameters ..... 51
  - 2.15.1 load\_Class\_Name - load\_Class\_Name: Name of the Data-Provider class ..... 51
  - 2.15.2 java\_class\_path - Java Class path ..... 51
  - 2.15.3 buffer\_size - Max size of transferred data block ..... 52
- 2.16 Global Parameters ..... 52
  - 2.16.1 buffer\_size - Max size of transferred data block ..... 52
  - 2.16.2 log\_filename - Name (and Pathname) of the logging-file ..... 52
  - 2.16.3 log4jPropertyFileName - Name (and Pathname) of a log4j configuration files ..... 53
  - 2.16.4 verbose - The granularity of (Debug-)Messages ..... 53
- 2.17 Params of the JobScheduler JADE-Jobs ..... 53
  - 2.17.1 Connection parameters ..... 53
    - 2.17.1.1 jump\_host ..... 53
    - 2.17.1.2 jump\_port ..... 53

- 2.17.1.3 jump\_protocol ..... 54
- 2.17.1.4 proxy parameters ..... 54
- 2.17.2 Authentication parameters ..... 54
  - 2.17.2.1 jump\_user - User name for authentication with the jump host ..... 54
  - 2.17.2.2 jump\_password ..... 54
  - 2.17.2.3 proxy parameters ..... 54
  - 2.17.2.4 jump\_proxy\_password ..... 55
  - 2.17.2.5 jump\_proxy\_user ..... 55
- 2.17.3 shell command parameters ..... 55
  - 2.17.3.1 jump\_command - shell command string ..... 55
  - 2.17.3.2 jump\_command\_delimiter - Delimiter character/string for multiple shell commands ..... 55
  - 2.17.3.3 jump\_command\_script - Name of executable script file ..... 56
  - 2.17.3.4 jump\_command\_script\_file ..... 56
- 2.17.4 Secure Shell (SSH) parameters ..... 56
  - 2.17.4.1 jump\_ssh\_auth\_file ..... 56
  - 2.17.4.2 jump\_ssh\_auth\_method ..... 56
  - 2.17.4.3 jump\_ignore\_error ..... 56
  - 2.17.4.4 jump\_ignore\_signal ..... 57
  - 2.17.4.5 jump\_ignore\_stderr ..... 57
  - 2.17.4.6 jump\_simulate\_shell ..... 57
  - 2.17.4.7 jump\_simulate\_shell\_inactivity\_timeout ..... 57
  - 2.17.4.8 jump\_simulate\_shell\_login\_timeout ..... 57
  - 2.17.4.9 jump\_simulate\_shell\_prompt\_trigger ..... 57
- 2.18 eMail Notification Parameter ..... 58
  - 2.18.1 mail\_on\_success - Send a Mail on Success ..... 58
  - 2.18.2 mail\_on\_error - Send Mail in case of Error ..... 58
  - 2.18.3 mail\_on\_empty\_files - eMail if empty files detected ..... 59
  - 2.18.4 ftp\_file\_notification\_to - Recipient of a notification mail ..... 59
  - 2.18.5 ftp\_file\_notification\_cc - eMail Notification: CC to ..... 59
  - 2.18.6 ftp\_file\_notification\_bcc - Bcc eMail Recipient ..... 59
  - 2.18.7 ftp\_file\_notification\_subject - Subject of an eMail Notification ..... 59
  - 2.18.8 ftp\_file\_notification\_body - eMail Body ..... 59
- 3 JobScheduler JADE-Job Parameters ..... 60**
  - 3.1 File Orders ..... 60
    - 3.1.1 File Order Creation ..... 60
      - 3.1.1.1 create\_order - Activate file-order creation ..... 60
      - 3.1.1.2 create\_orders\_for\_all\_files - Create a file-order for every file in the result-list ..... 60
      - 3.1.1.3 order\_jobchain\_name - The name of the jobchain which belongs to the order ..... 61
      - 3.1.1.4 next\_state - The first node to execute in a jobchain ..... 61
      - 3.1.1.5 merge\_order\_parameter - merge actual order parameter into new created order ..... 61
    - 3.1.2 File Order Parameter ..... 61
      - 3.1.2.1 scheduler\_file\_path - file to process for a file-order ..... 62
      - 3.1.2.2 scheduler\_file\_parent - pathname of the file to process for a file-order ..... 62
      - 3.1.2.3 scheduler\_file\_name - Name of the file to process for a file-order ..... 62
  - 3.2 Result Set ..... 62
    - 3.2.1 on\_empty\_result\_set - Set next node on empty result set ..... 62
    - 3.2.2 expected\_size\_of\_result\_set - number of expected hits in result-list ..... 63
    - 3.2.3 raise\_error\_if\_result\_set\_is - raise error on expected size of result-set ..... 63
    - 3.2.4 result\_list\_file - Name of the result-list file ..... 63
  - 3.3 Return Parameter ..... 63
    - 3.3.1 Parameters ..... 63
      - 3.3.1.1 scheduler\_sosfileoperations\_resultset - The result of the operation as a list of items ..... 64

---

3.3.1.2 scheduler_sosfileoperations_resultsetsize - The amount of hits in the result set of the operation . . .	64
3.3.1.3 scheduler_sosfileoperations_file_count - Return the size of the result set after a file operation . . . . .	64
<b>4 Index</b> .....	<b>65</b>

# 1 Introduction

Work in progress



## 2 JADE-Client Parameters

### JADE Client Parameters

#### 2.1 Transfer Profile Parameters

##### Client Parameters

##### 2.1.1 settings - Name of INI-File which contains the profiles to execute

A configuration (INI-) file can be specified that contains profiles, i.e. sections, with parameters specified as pairs of names and values in a plain text format like this:

```
[sample_transfer]
protocol = ftp
host = localhost
port = 21
source_dir = /tmp
...
```

##### "sample\_transfer" Profile

This is an example profile definition. The profile name is surrounded by square brackets. The name-value pairs are below the name of the profile. Each pair in a separate Line.

At the command line the name of the configuration file and the profile are specified like this:

```
jade.sh -settings=settings.ini -profile=sample_transfer ...
jade.cmd -settings=settings.ini -profile=sample_transfer ...
```

##### Setting the profile configuration to start the JADE client

Starting the JADE-Client at the command line and specifying names of the configuration file and Profile as parameters.

A profile can reference the contents of other profiles like this:

```
[default]
history = /jade/transfer_history.csv
mandator = SOS
scheduler_host = localhost
scheduler_port = 4444

[sample_transfer]
include = default
protocol = ftp
host = vulcan.sos
port = 21
source_dir = /tmp
...
```

##### Example for the include directive

With this example the file transfer history settings are specified in the the *default* profile. The *include* directive is used to call this profile from the *sample\_transfer* profile.

It is possible to include as many profiles as required with a single include directive, with the names of the profiles being separated by a comma.

The values of parameters with the same names will be overwritten, with the last parameter winning.

Data-Type: SOSOptionIniFileName

The default value for this parameter is **./Settings.ini**.

### 2.1.2 profile - The Name of a Profile-Section to be executed

If a configuration file is being used (see parameter [settings](#)) then this parameter specifies a name of a section within the configuration file. Such sections, i.e. profiles, specify parameters as pairs of names and values that otherwise would be specified by command line parameters.

At the command line the name of the configuration file and the profile are specified like this:

```
jade.sh -settings=settings.ini -profile=sample_transfer ...
jade.cmd -settings=settings.ini -profile=sample_transfer ...
```

Setting the profile configuration to start the JADE client

Starting the JADE-Client at the command line and specifying names of the configuration file and Profile as parameters.

Data-Type: SOSOptionString

### 2.1.3 operation - Operation to be executed

*send, receive, remove, execute or install.* send - transfer files by FTP/SFTP to a remote server receive - transfer files by FTP/SFTP from a remote server remove - remove files by FTP/SFTP on a remote server execute - execute a command by SSH on a remote server install - install SOSFTP on a remote server

Data-Type: SOSOptionStringValueList

The default value for this parameter is **send**.

This parameter is mandatory.

### 2.1.4 verbose - The granularity of (Debug-)Messages

Change granularity to granularity The *verbosity level* specifies the intensity of log entries. A value between 1 and 9 can be specified. Higher values cause more detailed information to be logged. Log output is written to stdout or to a file that has been specified with the parameter log\_filename.

Data-Type: SOSOptionInteger

The default value for this parameter is **10**.

### 2.1.5 protocol - Type of requested Datatransfer

The values ftp, sftp or ftps are valid for this parameter. If sftp is used, then the ssh\_\* parameters will be applied.

Data-Type: SOSOptionStringValueList

The default value for this parameter is **ftp**.

This parameter is mandatory.

## 2.2 Connection Parameters

Verbindungs-Parameter können mit dem Präfix "source\_" oder "target\_" feiner spezifiziert werden.

### 2.2.1 host - Host-Name

This parameter specifies the hostname (e.g. wilma.sos) or IP address (e.g. IPv4 192.168.0.1) of the (FTP, SFTP, SSH, SMTP, etc.) server to which a connection has to be made.

Data-Type: SOSOptionHostName

This parameter is mandatory.

### 2.2.2 port - Port-Number to be used for Data-Transfer

Port by which files should be transferred. For FTP this is usually port 21, for SFTP this is usually port 22.

Data-Type: SOSOptionPortNumber

The default value for this parameter is **21**.

This parameter is mandatory.

### 2.2.3 protocol - Type of requested Datatransfer

The values ftp, sftp or ftps are valid for this parameter. If sftp is used, then the ssh\_\* parameters will be applied.

Data-Type: SOSOptionStringValueList

The default value for this parameter is **ftp**.

This parameter is mandatory.

### 2.2.4 proxy parameters

#### 2.2.4.1 alternative\_host

Alternative parameter for the primary parameter host.

Data-Type: SOSOptionHostName

#### 2.2.4.2 alternative\_port

Alternative parameter for the primary parameter port.

Data-Type: SOSOptionPortNumber

### 2.3 Authentication Parameters

Authentication Parameters

#### 2.3.1 user - UserID of user in charge

User name for authentication at the (FTP/SFTP) server.

Data-Type: SOSOptionUserName

The default value for this parameter is **anonymous**.

This parameter is mandatory.

#### 2.3.2 password - Password for UserID

Password for authentication at the FTP/SFTP server. For SSH/SFTP connections that make use of public/private key authentication the *password* parameter is specified for the passphrase that optionally secures a private key.

Data-Type: SOSOptionPassword

#### 2.3.3 Alternative Authentication,

##### 2.3.3.1 alternative\_user

Alternative parameter for the primary parameter user .

Data-Type: SOSOptionUserName

The default value for this parameter is **anonymous**.

##### 2.3.3.2 alternative\_password

Alternative parameter for the primary parameter password .

Data-Type:

## 2.3.4 SSH only Parameters

### Client Parameters

#### 2.3.4.1 auth\_file

The value of this parameter specifies the path and name of a user's private key file used for authorisation on an SSH server. This parameter must be specified if the *publickey* authorization method has been specified in the *auth\_method* parameter. Should the private key file be secured by a password, then this must be specified using the *password* parameter.

Data-Type:

#### 2.3.4.2 auth\_method

Data-Type:

#### 2.3.4.3 strict\_hostkey\_checking - SSH HostKey checking

If this parameter is set to "yes", the application will not automatically add host keys to the `$HOME/.ssh/known_hosts` file, and will refuse to connect to hosts whose host key has changed.

This parameter forces the user to manually add all new hosts.

If this property is set to "no", the application will automatically add a new host key to the user known hosts files.

If this parameter is set to "ask", new host keys will be added to the user known host files only after the user has confirmed that is what they really want to do, and the application will refuse to connect to hosts whose host key has changed.

Setting this parameter to "yes" provides maximum protection against Trojan horse attacks.

See also <http://www.online-tutorials.net/security/secure-ssh-tutorial-part-1-host-key/tutorials-t-69-201.html> .

Data-Type: SOSOptionString

The default value for this parameter is **no**.

#### 2.3.4.4 SSH proxy parameter

##### 2.3.4.4.1 ssh\_proxy\_host

The value of this parameter is the host name or the IP address of a proxy that is used in order to establish a connection to the SSH server. The use of a proxy is optional.

Data-Type:

#### 2.3.4.4.2 ssh\_proxy\_port - Port number of proxy

This parameter specifies the port number of the proxy, should a proxy be used in order to establish a connection, e.g. to a SSH server.

Data-Type:

#### 2.3.4.4.3 ssh\_proxy\_user - User-ID for the Proxy-Server

The value of this parameter specifies the user account for authentication by the proxy server should a proxy be used in order to connect to the SSH server.

Data-Type:

#### 2.3.4.4.4 ssh\_proxy\_password - Password of Proxy-Server

This parameter specifies the password for the proxy server user account, should a proxy be used in order to connect to the SSH server.

Data-Type:

## 2.4 Selecting Objects to Transfer

### Client Parameters

The options that can be used to select the object which is to be transferred are described in this chapter.

#### 2.4.1 file\_path

This parameter is used to specify an individual file or files for transfer. It is an alternative to the file\_spec parameter. More than one file can be specified by using a ";" between the file names.

All files will be processed in the order they are defined in this parameter.

The following applies when files are being received:

This parameter accepts the absolute name and path of file at the FTP/SFTP server that should be transferred. The file name has to include both name and path of the file at the FTP/SFTP server.

The file will be stored under its name in the directory that is specified by the local\_dir parameter.

The following parameters are ignored should this parameter be used:

file\_spec and

remote\_dir.

The following applies when files are being sent:

This parameter accepts the absolute name and path of file that should be transferred. An absolute path has to be specified.

The file will be stored under its name in the directory at the FTP/SFTP server that has been specified by the `remote_dir` parameter.

The following parameters are ignored should this parameter be used:

`file_spec` and

`local_dir`.

Data-Type: `SOSOptionString`

### 2.4.2 `file_spec` - File filtering using a regular expression

This parameter specifies a regular expression, which is used to select files from a directory. All file names that correspond to the regular expression are transferred to a results list. All files whose names are contained in this list will be transferred.

Data-Type: `SOSOptionRegExp`

The default value for this parameter is `.*`.

### 2.4.3 `file_list_name` - Name of a file which contains a filelist

Each record in this file contains a file name which has to be processed. All files in the filelist will be processed in the the sequence they appear in the file.

If a file in the list does not exist the processing will be aborted.

```
/home/test/abc.de  
/inbound/daily/cde.dat  
/work/hello.world
```

**Example: a filelist**

The filelist contains three records. Each record contains a filename which has to be processed.

Data-Type: `SOSOptionInFileName`

The default value for this parameter is `.`

## 2.5 Transactional Transfer

Transactional Transfer

### 2.5.1 `transactional` - Transaction controlled Transfer

This parameter specifies whether a transfer should be processed within a single transaction, i.e. either all objects are successfully transferred or none. Should an error occur during a transfer operation then all transfers will be rolled back.

When specifying the value `true` then the following applies:

The parameter `atomic_suffix` or `atomic_prefix` has to be specified that causes target files to be created with a suffix such as "~" and that causes the respective files to be renamed to their target file name after the transfer of all files has been successfully completed. If at least one file out of a set of files cannot be transferred successfully then no files will be renamed, instead the temporarily created files are removed from the target system.

The parameter `remove_files` that causes files to be removed after successful transfer will be effective only after all files have been successfully transferred. Otherwise no files will be removed.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [atomic\\_prefix](#), [atomic\\_suffix](#)

### 2.5.2 `atomic_suffix` - Suffix for the target transfer file name

This parameter specifies whether target files should be created with a suffix such as "~", and should be renamed to the target file name after the file transfer is completed. This mechanism is useful if the target directory is monitored for incoming files by some application and if files are required to appear atomically instead of being subsequently written to.

The temporary suffix is specified as the value of this parameter.

This setting is recommended should target directories be monitored by an application or the JobScheduler.

Data-Type: SOSOptionString

### 2.5.3 `atomic_prefix` - Prefix for the target transfer file name

This parameter specifies whether target files should be created with a prefix such as "~" and have to be renamed to the target file name after the file transfer has been completed without errors. This mechanism is useful if the target directory is monitored for incoming files by an application such as JobScheduler and if files are required to appear atomically - i.e during transfer - instead of being shown after transfer has been completed.

The value of this parameter is the temporary prefix.

Data-Type: SOSOptionString

## 2.6 Atomic Transfer

One can define a Temporary File Prefix or Suffix, such as "~", to be used as a part of the target transfer filename until the completion of a successful transfer. JADE will create a target transfer filename as a combination of the original source file name (or the source transfer filename) and the prefix and/or the suffix.

This functionality protects the objects during the transfer, e.g. prevents the processing of the objects by a monitoring program.

This prefix will be added to the beginning of the name, the suffix to the end, of the target transfer file to be transferred, until the transfer is successfully completed. This ensures that even if the target file is of the same name



and the transfer profile is configured to overwrite such files, that file will not be overwritten until the file transfer is successful completed.

### 2.6.1 atomic\_suffix - Suffix for the target transfer file name

This parameter specifies whether target files should be created with a suffix such as "~", and should be renamed to the target file name after the file transfer is completed. This mechanism is useful if the target directory is monitored for incoming files by some application and if files are required to appear atomically instead of being subsequently written to.

The temporary suffix is specified as the value of this parameter.

This setting is recommended should target directories be monitored by an application or the JobScheduler.

Data-Type: SOSOptionString

### 2.6.2 atomic\_prefix - Prefix for the target transfer file name

This parameter specifies whether target files should be created with a prefix such as "~" and have to be renamed to the target file name after the file transfer has been completed without errors. This mechanism is useful if the target directory is monitored for incoming files by an application such as JobScheduler and if files are required to appear atomically - i.e during transfer - instead of being shown after transfer has been completed.

The value of this parameter is the temporary prefix.

Data-Type: SOSOptionString

## 2.7 Operations on Objects

Operations on Objects

### 2.7.1 Concurrent Transfer Parameters

Concurrent Transfer Parameters

#### 2.7.1.1 concurrent\_transfer - Activate parallel transfers

Data-Type: SOSOptionString

The default value for this parameter is **false**.

Use together with parameter: [max\\_concurrent\\_transfers](#)

### 2.7.1.2 max\_concurrent\_transfers - Max number of parallel transfers

Data-Type: SOSOptionInteger

The default value for this parameter is **3**.

Use together with parameter: [concurrent\\_transfer](#)

## 2.7.2 Compressing Files

Compressing Files

### 2.7.2.1 compress\_files

This parameter specifies whether or not the content of the source files should be compressed by using a zip-algorithm.

In case of sending files each files which has to be selected for sending will be compressed in a single zip file. The extension of the file-name is defined with the parameter compress\_filename\_extension.

A gzip-compatible compression is used, no further software components are required.

If compress\_files is used the parameter append\_files is not allowed.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.7.2.2 compressed\_file\_extension - Additional file-name extension for compressed files

This parameter specifies the file extension should target file compression be specified using the compress\_files parameter.

Data-Type: SOSOptionString

The default value for this parameter is **.gz**.

### 2.7.2.3 decompress\_after\_transfer - Unpack Files after Transfer

Data-Type: SOSOptionString

The default value for this parameter is **false**.

## 2.7.3 Cumulate File Parameters

With the options described in this chapter one can be determined that the files from the resultlist not should be transferred separately, but are to be combined prior to transfer to another, further file (target file) consecutively.

The name of the file is set with the parameter cumulative\_filename .

With the parameter `cumulative_file_separator` is set a text that will be inserted between the individual files in the target file.

The files can be deleted after transfer. This is determined with the parameter `cumulative_file_delete` .

### 2.7.3.1 `cumulate_files` - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.7.3.2 `cumulative_filename` - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionOutFileName

The default value for this parameter is .

Use together with parameter: [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

### 2.7.3.3 `cumulative_file_separator` - cumulate (all) files into a single file

This option causes files from the results list to be cumulated together into a single target file, one after the other, before being transferred. This is in contrast to the default transfer mode in which files are transferred individually.

The name of the cumulative file is set using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

#### 2.7.3.4 cumulative\_file\_delete - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

### 2.7.4 Changing File Names

The possibilities for changing object (file) names are described in this chapter. File names can be changed both at the source and at the target.

#### 2.7.4.1 replacing - Regular expression for renaming Files

This parameter has only effect on the target files. You can also use target\_replacing as parameter name. If you want to rename source files then use source\_replacing.

Regular expression for filename replacement with the value of the parameter replacement .

If the expression matches the filename then the groups found are replaced.

a) For replacement "capturing groups" are used. Only the content of the capturing groups is replaced.

Replacements are separated by a semicolon ";". Example:

```
replacing= (1)abc(12)def(.*)
```

```
replacement = A;BB;CCC
```

```
Name of original file: 1abc12def123.txt
```

```
Name after renaming: AabcBBdefCCC
```

b) If no "capturing groups" are specified then the entire match is replaced. Example:

```
replacing = Hello
```

replacement = 1234

Input file: Hello\_World.txt

Output file: 1234\_World.txt

Requires the parameter replacement to be specified.

**Examples for regular expressions used for replacing:**

source_file	file_spec	replacing	replacement	Example Input	Output	Comment
./temp	.*	~	""	hallo.dat~	hallo.dat	-
a.txt	-	[.]txt	_2011-11-24.txt	-	a_2011-11-24.txt	-
./temp	.*	.*	[filename:] [date:yyyy-MM-dd].dat	a.txt	a.txt2011-11-24.dat	[filename:] and [date:] will be substituted after file has been replaced
./temp	*[.]txt	a.*?b	world	a.1.b.txt.b	world.txt.b	*? : ungreedy usage of .*
abcdefg.txt	-	^(ab)cde(fg)[.]txt	123:[date:yyyy-MM-dd]	-	123cde2011-11-24.txt	; : separator between multiple replacements
./temp	*.dat\$	(ab)_c[0-9]d_(fg)	group1;group2	ab_c1d_fg. ab_c2d_fg.dat	group1_c1d_group2. ab_c2d_fg.dat	Only the first pattern match is replaced
./temp	.*	^(?:aa bb)_{2}_[0-9]+_[0-9]+\$	tail	aa_QU_061205_222334	aa_QU_tail	(?:...): "?" specifies a non capturing group. The file will not be replaced.
./temp	FILE.txt	.*	[filename:lowercase]_[filename:uppercase]_[filename:]	FILE.txt	file.txt_FILE.TXT_FILE.txt	-

source_file	target_file	file_spec	create_dir	replacing	replacement	Example input	Output	Comment
aaa/bbb/	[directory:]/ [date:yyyyMMd]/	hello.txt	true	[.]txt	[date:yyyy].txt	aaa/bbb/hello.txt	aaa/bbb/20070316/hello2007.txt	If the directory aaa/bbb/20070316 does not yet exist it will be created

For further information see [java.util.regex.Pattern](#)

Data-Type: SOSOptionRegExp

Use together with parameter: [replacement](#)

#### 2.7.4.2 replacement - String for renaming Filenames

This parameter has only effect on the target files. You can also use `target_replacement` as parameter name. If you want to rename source files then use `source_replacement`.

String for replacement of matching character sequences within file names that are specified with the value of the parameter `replacing`.

If multiple "capturing groups" are be replaced then one replacement string per group has to be specified. These strings are separated by a semicolon ";":

replacement: aa;[filename:];bb

Supports masks for substitution in the file name with format strings that are enclosed with `[` and `]`. The following format strings are supported:

[date: date format ]

*date format* must be a valid Java data format string, e.g. `yyyyMMddHHmmss` , `yyyy-MM-dd.HHmmss` etc.

[date: date format ] *date format* must be a valid Java data format string, e.g. `yyyyMMddHHmmss` , `yyyy-MM-dd.HHmmss` etc. [filename:] Will be substituted by the original file name including the file extension. [filename:lowercase] Will be substituted by the original file name including the file extension with all characters converted to lower case. [filename:uppercase] Will be substituted by the original file name including the file extension with all characters converted to upper case.

Requires the parameter `replacing` to be specified.

Data-Type: SOSOptionString

Use together with parameter: [replacing](#)

#### 2.7.4.3 Examples: Cumulate Files

### 2.7.4.3.1 Create Subfolder with timestamp

```
[CopyAndCreateVariableFolder_Local2Local]

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SAVE[date:yyyyMMddHHmm]/
make_Dirs = true

file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
remove_files = false
@Test
public void testCopyAndCreateVariableFolder() throws Exception {
    setSourceAndTarget();

    objOptions.FileNamePatternRegExp.Value("^.*\.txt$");
    objOptions.operation.Value(enuJadeOperations.copy);

    objOptions.TargetDir.Value(strTestPathName + "/SAVE[date:yyyyMMddHHmm]");
    objOptions.makeDirs.value(true);
    objOptions.remove_files.value(false);

    startTransfer(objOptions);
}
... ./jade.sh -settings=settings-file-name -profile=CopyAndCreateVariableFolder_Local2Local
```

Rename files on source and target

### 2.7.4.3.2

```
[CopyAndMoveSource_Local2Local]

source_replacing=(.*)\.txt)
source_replacement=/SAVE/\1_[date:yyyyMMddHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
@Test
public void testCopyAndRenameSource() throws Exception {
    setSourceAndTarget();

    objOptions.FileNamePatternRegExp.Value("^.*\.txt$");
    objOptions.operation.Value(enuJadeOperations.copy);

    objOptions.Source().replacing.Value("(.*)\.txt");
    objOptions.Source().replacement.Value("/SAVE/\1_[date:yyyyMMddHHmm];\2");

    objOptions.remove_files.value(false);

    startTransfer(objOptions);
}
jade.sh -settings=settings-file-name -profile=CopyAndMoveSource_Local2Local
```

Rename files on source and target

### 2.7.4.3.3

```
[CopyAndRenameSource_Local2Local]
source_replacing=(.*)\.txt)
source_replacement=\1_[date:yyyyMMddHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
```

Rename files on source and target

### 2.7.4.3.4

```
[CopyAndRenameSourceAndTarget_Local2Local]
source_replacing = (.*).txt)
source_replacement=\1_[date:yyyyMMddHHmm];\2

replacement      = \1_[date:yyyyMMdd];\2
replacing        = (.*).txt)

source_dir       = J:/E/java/junittests/testdata/JADE/
target_dir       = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec        = ^.*\.txt$
operation        = copy
source_protocol  = local
target_protocol  = local
remove_files     = false
```

Rename files on source and target

## 2.7.5 Encoding

Encoding

### 2.7.5.1 file\_name\_encoding - Set the encoding-type of a file name

Data-Type: SOSOptionString

The default value for this parameter is .

### 2.7.5.2 control\_encoding - Specify the encoding-type used by the server

Data-Type: SOSOptionString

The default value for this parameter is .



## 2.8 Data Source only Parameters

### Data Source only Parameters

#### 2.8.1 force\_files - Raise Error, if no files found for transfer

This parameter specifies whether an error should be raised if no files could be found for transfer. If the value is *false* no error is reported.

The number of files to be transferred is determined by the `file_spec` or `file_path` parameters and can be restricted by the `overwrite_files` parameter should this be specified with the value *false* .

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **true**.

#### 2.8.2 recursive - Include subfolders in operation

This parameter specifies whether files from subdirectories should be included recursively.

Only effective if `file` is a directory.

Regular expression matches apply to files from subdirectories as specified by the parameter `file_spec` .

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.8.3 local\_dir - local\_dir

Local directory into which or from which files should be transferred. By default the current working directory is used.

Besides paths in the local file system UNC path names are supported that could be used to address remote server systems: `\\somehost\somedirectory` can be used in the same way as `//somehost/somedirectory` to transfer files from an FTP/SFTP server to a different remote server system.

Moreover, you could specify URIs for a file schema as in `file:///somehost/somedirectory` . Please, consider the required number of slashes. file URIs are subject to the following limitations due to constraints of the underlying Java JRE:

File names and path names must not contain any spaces. Authentication by authority strings as in `file:///user:password@somehost/somedirectory` is not supported.

Data-Type: SOSOptionFolderName

The default value for this parameter is `..`

This parameter is mandatory.

## 2.8.4 remove\_files

This parameter specifies whether files on the source server should be removed after transfer.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

## 2.8.5 zero\_byte\_transfer - Transfer of "zero byte" size files

Dateien mit 0 Byte Größe werden übertragen.

: Dateien mit 0 Byte Größe werden übertragen, wenn mindestens eine der übertragenen Dateien mehr als 0 Byte Größe aufweist.

Files with zero byte size are not transferred. An error will be raised if any zero byte file is found.

Files with zero byte size will not be transferred. However, no error will be raised if this results in no files being transferred.

This parameter specifies whether zero byte files should be transferred and processed by subsequent commands. The following settings are available:

yes Files with zero byte size are transferred (default). no Files with zero byte size are transferred, should at least one of the files have more than zero byte size. strict Files with zero byte size are not transferred. An error will be raised if any zero byte file is found. relaxed Files with zero byte size will not be transferred. However, no error will be raised if this results in no files being transferred.

Use of this parameter can be refined using the force\_files parameter: should force\_files have the value false , then processing will be treated as successful in the event of no files having been transferred.

Note that the remove\_files parameter has unrestricted validity. Files with zero byte size will be removed regardless of whether or not they have been transferred.

Data-Type: SOSOptionStringValueList

The default value for this parameter is **yes**.

## 2.8.6 Changing File Names

The possibilities for changing object (file) names are described in this chapter. File names can be changed both at the source and at the target.

### 2.8.6.1 replacing - Regular expression for renaming Files

This parameter has only effect on the target files. You can also use target\_replacing as parameter name. If you want to rename source files then use source\_replacing.

Regular expression for filename replacement with the value of the parameter replacement .

If the expression matches the filename then the groups found are replaced.

a) For replacement "capturing groups" are used. Only the content of the capturing groups is replaced.

Replacements are separated by a semicolon ";". Example:

replacing= (1)abc(12)def(.\*)

replacement = A;BB;CCC

Name of original file: 1abc12def123.txt

Name after renaming: AabcBBdefCCC

b) If no "capturing groups" are specified then the entire match is replaced. Example:

replacing = Hello

replacement = 1234

Input file: Hello\_World.txt

Output file: 1234\_World.txt

Requires the parameter replacement to be specified.

**Examples for regular expressions used for replacing:**

source_file	file_spec	replacing	replacement	Example Input	Output	Comment
./temp	.*	~	""	hallo.dat~	hallo.dat	-
a.txt	-	[.]txt	_2011-11-24.txt	-	a_2011-11-24.txt	-
./temp	.*	.*	[filename:] [date:yyyy-MM-dd].dat	a.txt	a.txt2011-11-24.dat	[filename:] and [date:] will be substituted after file has been replaced
./temp	*[.]txt	a.*?b	world	a.1.b.txt.b	world.txt.b	*? : ungreedy usage of *
abcdefg.txt	-	^(ab)cde(fg)[.]txt	123;[date:yyyy-MM-dd]	-	123cde2011-11-24.txt	; : separator between multiple replacements
./temp	*.dat\$	(ab)_c[0-9]d_(fg)	group1;group2	ab_c1d_fg. ab_c2d_fg.dat	group1_c1d_group2. ab_c2d_fg.dat	Only the first pattern match is replaced
./temp	.*	^(?:aa bb)_{2}_[0-9]+_[0-9]+)\$	tail	aa_QU_061205_222334	aa_QU_tail	(?:...): "?" specifies a non capturing group. The file will not be replaced.

./temp	FILE.txt	.*	[filename:lowercase]_[filename:uppercase]_[filename:]	FILE.txt	file.txt_FILE.TXT_FILE.txt	-
--------	----------	----	---	----------	----------------------------	---

source_file	target_file	file_spec	create_dir	replacing	replacement	Example input	Output	Comment
aaa/bbb/	[directory:]/ [date:yyyyMMdd]/	hello.txt	true	[.]txt	[date:yyyy].txt	aaa/bbb/hello.txt	aaa/bbb/20070316/hello2007.txt	If the directory aaa/bbb/20070316 does not yet exist it will be created

For further information see [java.util.regex.Pattern](#)

Data-Type: SOSOptionRegExp

Use together with parameter: [replacement](#)

### 2.8.6.2 replacement - String for renaming Filenames

This parameter has only effect on the target files. You can also use target\_replacement as parameter name. If you want to rename source files then use source\_replacement.

String for replacement of matching character sequences within file names that are specified with the value of the parameter replacing.

If multiple "capturing groups" are be replaced then one replacement string per group has to be specified. These strings are separated by a semicolon ";":

replacement: aa;[filename:];bb

Supports masks for substitution in the file name with format strings that are enclosed with [ and ]. The following format strings are supported:

[date: date format ]

*date format* must be a valid Java data format string, e.g. yyyyMMddHHmmss , yyyy-MM-dd.HHmmss etc.

[date: date format ] *date format* must be a valid Java data format string, e.g. yyyyMMddHHmmss , yyyy-MM-dd.HHmmss etc. [filename:] Will be substituted by the original file name including the file extension. [filename:lowercase] Will be substituted by the original file name including the file extension with all characters converted to lower case. [filename:uppercase] Will be substituted by the original file name including the file extension with all characters converted to upper case.

Requires the parameter replacing to be specified.

Data-Type: SOSOptionString

Use together with parameter: [replacing](#)

## 2.8.6.3 Examples: Cumulate Files

### 2.8.6.3.1 Create Subfolder with timestamp

```
[CopyAndCreateVariableFolder_Local2Local]

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SAVE[date:yyyyMMdHHmm]/
make_Dirs = true

file_spec = ^.*\..txt$
operation = copy
source_protocol = local
target_protocol = local
remove_files = false @Test
    public void testCopyAndCreateVariableFolder() throws Exception {
        setSourceAndTarget();

        objOptions.FileNamePatternRegExp.Value("^.*\..txt$");
        objOptions.operation.Value(enuJadeOperations.copy);

        objOptions.TargetDir.Value(strTestPathName + "/SAVE[date:yyyyMMdHHmm]");
        objOptions.makeDirs.value(true);
        objOptions.remove_files.value(false);

        startTransfer(objOptions);
    }
    ... ./jade.sh -settings=settings-file-name -profile=CopyAndCreateVariableFolder_Local2Local
```

Rename files on source and target

### 2.8.6.3.2

```
[CopyAndMoveSource_Local2Local]

source_replacing=(.*)(.txt)
source_replacement=/SAVE/\1_[date:yyyyMMdHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\..txt$
operation = copy
source_protocol = local
target_protocol = local
@Test
    public void testCopyAndRenameSource() throws Exception {
        setSourceAndTarget();

        objOptions.FileNamePatternRegExp.Value("^.*\..txt$");
        objOptions.operation.Value(enuJadeOperations.copy);

        objOptions.Source().replacing.Value("(.*)(.txt)");
        objOptions.Source().replacement.Value("/SAVE/\1_[date:yyyyMMdHHmm];\2");

        objOptions.remove_files.value(false);

        startTransfer(objOptions);
    }
    jade.sh -settings=settings-file-name -profile=CopyAndMoveSource_Local2Local
```

Rename files on source and target

### 2.8.6.3.3

```
[CopyAndRenameSource_Local2Local]
source_replacing=(.*)\.txt)
source_replacement=\1_[date:yyyyMMdHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
```

Rename files on source and target

### 2.8.6.3.4

```
[CopyAndRenameSourceAndTarget_Local2Local]
source_replacing = (.*).txt)
source_replacement=\1_[date:yyyyMMdHHmm];\2

replacement = \1_[date:yyyyMMd];\2
replacing = (.*).txt)

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
remove_files = false
```

Rename files on source and target

## 2.8.7 Check the completeness of a file (steady state)

List of parameters

Name	Title
<a href="#">check_steady_state_of_files</a>	
<a href="#">steady_state_count</a>	
<a href="#">check_steady_state_interval</a>	

### 2.8.7.1 check\_steady\_state\_of\_files - Check the completeness of a file (steady state)

In some file transfer scenarios has the receiver of a file no knowledge about the time when the sender creates the file. In case of a (very) large file it can be the situation that the receiver tries to read the file but the sender has not finished to write it. If the receiver get the file at the moment the sender is still writing, as a result he will get a corrupted, incomplete file.

Setting this parameter to "true" the receiver will check the file for completeness before he starts the transfer.

At the end, this is not a very secure approach, because the receiver is checking the date of last modification and the size of the file. If both not changing between a time intervall, which is defined by the parameters ..., the file is

guessed to be complete. If the sender is terminated without writing the complete file, or the network is down, or the speed of processing the file is going slow, the receiver will get a corrupted file.

A better approach for avoiding corrupt files is to use the atomic method: writing a file and after completion of writing rename the file. For more details about this method see parameter [atomic\\_suffix](#) or [atomic\\_prefix](#).

If more than one file is to be transferred, the transactional approach is the first choice. See parameter [transactional](#).

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [check\\_steady\\_state\\_interval](#), [steady\\_state\\_count](#)

### 2.8.7.2 steady\_state\_count - Maximum Number of Checkpoints

The value of this option specifies the number of retries for to check the steady state of a file.

Data-Type: SOSOptionInteger

The default value for this parameter is **30**.

Use together with parameter: [check\\_steady\\_state\\_interval](#), [check\\_steady\\_state\\_of\\_files](#)

### 2.8.7.3 check\_steady\_state\_interval - temporal distance between checkpoints

The temporal distances in seconds between two checkpoints is defined by the value of this option.

Data-Type: SOSOptionTime

The default value for this parameter is **1**.

Use together with parameter: [check\\_steady\\_state\\_of\\_files](#), [steady\\_state\\_count](#)

## 2.9 Data Target Parameters

Client Parameters

### 2.9.1 transactional - Transaction controlled Transfer

This parameter specifies whether a transfer should be processed within a single transaction, i.e. either all objects are successfully transferred or none. Should an error occur during a transfer operation then all transfers will be rolled back.

When specifying the value true then the following applies:

The parameter `atomic_suffix` or `atomic_prefix` has to be specified that causes target files to be created with a suffix such as "~" and that causes the respective files to be renamed to their target file name after the transfer of all files has been successfully completed. If at least one file out of a set of files cannot be transferred successfully then no files will be renamed, instead the temporarily created files are removed from the target system.

The parameter `remove_files` that causes files to be removed after successful transfer will be effective only after all files have been successfully transferred. Otherwise no files will be removed.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [atomic\\_prefix](#), [atomic\\_suffix](#)

### 2.9.2 `append_files` - Append content of transferred files to existing files

This parameter specifies whether the content of a source file should be appended to the target file, should the target file exist.

The parameter [overwrite\\_files](#) will be ignored if this parameter is specified with the value true.

A message of type "info" will be created.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.9.3 `overwrite_files` - Overwrite existing files

This parameter specifies whether existing files can be overwritten.

Is this parameter used with `force_files` and should no files be transferred due to overwrite protection then an error will be raised stating that "no matching files" could be found.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **true**.

### 2.9.4 `make_dirs` - Create folder

If enabled the folder, which is specified as target, is checked for existence. If the folder does not exist it will be created as a directory.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.



## 2.9.5 atomic\_prefix - Prefix for the target transfer file name

This parameter specifies whether target files should be created with a prefix such as "~" and have to be renamed to the target file name after the file transfer has been completed without errors. This mechanism is useful if the target directory is monitored for incoming files by an application such as JobScheduler and if files are required to appear atomically - i.e during transfer - instead of being shown after transfer has been completed.

The value of this parameter is the temporary prefix.

Data-Type: SOSOptionString

## 2.9.6 atomic\_suffix - Suffix for the target transfer file name

This parameter specifies whether target files should be created with a suffix such as "~", and should be renamed to the target file name after the file transfer is completed. This mechanism is useful if the target directory is monitored for incoming files by some application and if files are required to appear atomically instead of being subsequently written to.

The temporary suffix is specified as the value of this parameter.

This setting is recommended should target directories be monitored by an application or the JobScheduler.

Data-Type: SOSOptionString

## 2.9.7 remote\_dir - Name of the Target Folder

Directory at the FTP/SFTP server from which or to which files should be transferred.

By default the home directory of the user at the FTP/SFTP server is used.

Data-Type: SOSOptionFolderName

The default value for this parameter is ..

This parameter is mandatory.

## 2.9.8 alternative\_remote\_dir

Alternative parameter for the primary parameter remote\_dir.

Data-Type:

## 2.9.9 strict\_hostkey\_checking - SSH HostKey checking

If this parameter is set to "yes", the application will not automatically add host keys to the \$HOME/.ssh/known\_hosts file, and will refuse to connect to hosts whose host key has changed.

This parameter forces the user to manually add all new hosts.

If this property is set to "no", the application will automatically add a new host key to the user known hosts files.

If this parameter is set to "ask", new host keys will be added to the user known host files only after the user has confirmed that is what they really want to do, and the application will refuse to connect to hosts whose host key has changed.

Setting this parameter to "yes" provides maximum protection against Trojan horse attacks.

See also <http://www.online-tutorials.net/security/secure-ssh-tutorial-part-1-host-key/tutorials-t-69-201.html> .

Data-Type: SOSOptionString

The default value for this parameter is **no**.

## 2.9.10 Changing File Names

The possibilities for changing object (file) names are described in this chapter. File names can be changed both at the source and at the target.

### 2.9.10.1 replacing - Regular expression for renaming Files

This parameter has only effect on the target files. You can also use `target_replacing` as parameter name. If you want to rename source files then use `source_replacing`.

Regular expression for filename replacement with the value of the parameter replacement .

If the expression matches the filename then the groups found are replaced.

a) For replacement "capturing groups" are used. Only the content of the capturing groups is replaced.

Replacements are separated by a semicolon ";". Example:

```
replacing= (1)abc(12)def(.*)
```

```
replacement = A;BB;CCC
```

Name of original file: 1abc12def123.txt

Name after renaming: AabcBBdefCCC

b) If no "capturing groups" are specified then the entire match is replaced. Example:

```
replacing = Hello
```

```
replacement = 1234
```

Input file: Hello\_World.txt

Output file: 1234\_World.txt

Requires the parameter replacement to be specified.

#### Examples for regular expressions used for replacing:

source_file	file_spec	replacing	replacement	Example Input	Output	Comment
-------------	-----------	-----------	-------------	---------------	--------	---------

./temp	.*	~	""	hallo.dat~	hallo.dat	-
a.txt	-	[.]txt	_2011-11-24.txt	-	a_2011-11-24.txt	-
./temp	.*	.*	[filename:] [date:yyyy-MM-dd].dat	a.txt	a.txt2011-11-24.dat	[filename:] and [date:] will be substituted after file has been replaced
./temp	*[.]txt	a.*?b	world	a.1.b.txt.b	world.txt.b	*? : ungreedy usage of .*
abcdefg.txt	-	^(ab)cde(fg)[.]txt	123;[date:yyyy-MM-dd]	-	123cde2011-11-24.txt	; : separator between multiple replacements
./temp	*.dat\$	(ab)_c[0-9]d_(fg)	group1;group2	ab_c1d_fg. ab_c2d_fg.dat	group1_c1d_group2. ab_c2d_fg.dat	Only the first pattern match is replaced
./temp	.*	^(?:aa bb){2}_[0-9]+_[0-9]+\$	tail	aa_QU_061205_222334	aa_QU_tail	(?:...): "?" specifies a non capturing group. The file will not be replaced.
./temp	FILE.txt	.*	[filename:lowercase]_[filename:uppercase]_[filename:]	FILE.txt	file.txt_FILE.TXT_FILE.txt	-

source_file	target_file	file_spec	create_dir	replacing	replacement	Example input	Output	Comment
aaa/bbb/	[directory:]/ [date:yyyyMMdd]/	hello.txt	true	[.]txt	[date:yyyy].txt	aaa/bbb/hello.txt	aaa/bbb/20070316/hello.txt	If the directory aaa/bbb/20070316 does not yet exist it will be created

For further information see [java.util.regex.Pattern](http://java.util.regex.Pattern)

Data-Type: SOSOptionRegExp

Use together with parameter: [replacement](#)

### 2.9.10.2 replacement - String for renaming Filenames

This parameter has only effect on the target files. You can also use target\_replacement as parameter name. If you want to rename source files then use source\_replacement.

String for replacement of matching character sequences within file names that are specified with the value of the parameter replacing.

If multiple "capturing groups" are be replaced then one replacement string per group has to be specified. These strings are separated by a semicolon ";":

replacement: aa;[filename:];bb

Supports masks for substitution in the file name with format strings that are enclosed with [ and ] . The following format strings are supported:

[date: date format ]

*date format* must be a valid Java data format string, e.g. *yyyyMMddHHmmss* , *yyyy-MM-dd.HHmmss* etc.

[date: date format ] *date format* must be a valid Java data format string, e.g. *yyyyMMddHHmmss* , *yyyy-MM-dd.HHmmss* etc. [filename:] Will be substituted by the original file name including the file extension. [filename:lowercase] Will be substituted by the original file name including the file extension with all characters converted to lower case. [filename:uppercase] Will be substituted by the original file name including the file extension with all characters converted to upper case.

Requires the parameter replacing to be specified.

Data-Type: SOSOptionString

Use together with parameter: [replacing](#)

### 2.9.10.3 Examples: Cumulate Files

#### 2.9.10.3.1 Create Subfolder with timestamp

```
[CopyAndCreateVariableFolder_Local2Local]

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SAVE[date:yyyyMMddHHmm]/
make_Dirs = true

file_spec = ^.*\..txt$
operation = copy
source_protocol = local
target_protocol = local
remove_files = false @Test
public void testCopyAndCreateVariableFolder() throws Exception {
    setSourceAndTarget();

    objOptions.FileNamePatternRegExp.Value("^.*\..txt$");
    objOptions.operation.Value(enuJadeOperations.copy);

    objOptions.TargetDir.Value(strTestPathName + "/SAVE[date:yyyyMMddHHmm]");
    objOptions.makeDirs.value(true);
    objOptions.remove_files.value(false);

    startTransfer(objOptions);
}
... ./jade.sh -settings=settings-file-name -profile=CopyAndCreateVariableFolder_Local2Local
```

Rename files on source and target

## 2.9.10.3.2

```
[CopyAndMoveSource_Local2Local]

source_replacing=(.*)(.txt)
source_replacement=/SAVE/\1_[date:yyyyMMddHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
@Test
public void testCopyAndRenameSource() throws Exception {
    setSourceAndTarget();

    objOptions.FileNamePatternRegExp.Value("^.*\\.txt$");
    objOptions.operation.Value(enuJadeOperations.copy);

    objOptions.Source().replacing.Value("(.*)(.txt)");
    objOptions.Source().replacement.Value("/SAVE/\1_[date:yyyyMMddHHmm];\2");

    objOptions.remove_files.value(false);

    startTransfer(objOptions);
}
jade.sh -settings=settings-file-name -profile=CopyAndMoveSource_Local2Local
```

Rename files on source and target

## 2.9.10.3.3

```
[CopyAndRenameSource_Local2Local]

source_replacing=(.*)(.txt)
source_replacement=\1_[date:yyyyMMddHHmm];\2

source_dir = J:/E/java/junittests/testdata/JADE/
target_dir = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec = ^.*\.txt$
operation = copy
source_protocol = local
target_protocol = local
```

Rename files on source and target

## 2.9.10.3.4

```
[CopyAndRenameSourceAndTarget_Local2Local]
source_replacing =(.*)(.txt)
source_replacement=\1_[date:yyyyMMddHHmm];\2

replacement      = \1_[date:yyyyMMdd];\2
replacing         = (.*)(.txt)

source_dir       = J:/E/java/junittests/testdata/JADE/
target_dir       = J:/E/java/junittests/testdata/JADE/SOSMDX/
file_spec        = ^.*\.txt$
operation        = copy
source_protocol  = local
target_protocol  = local
remove_files     = false
```

Rename files on source and target

## 2.9.11 Cumulate File Parameters

With the options described in this chapter one can be determined that the files from the resultlist not should be transferred separately, but are to be combined prior to transfer to another, further file (target file) consecutively.

The name of the file is set with the parameter `cumulative_filename` .

With the parameter `cumulative_file_separator` is set a text that will be inserted between the individual files in the target file.

The files can be deleted after transfer. This is determined with the parameter `cumulative_file_delete` .

### 2.9.11.1 `cumulate_files` - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.9.11.2 `cumulative_filename` - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionOutFileName

The default value for this parameter is .

Use together with parameter: [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

### 2.9.11.3 `cumulative_file_separator` - cumulate (all) files into a single file

This option causes files from the results list to be cumulated together into a single target file, one after the other, before being transferred. This is in contrast to the default transfer mode in which files are transferred individually.

The name of the cumulative file is set using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

#### 2.9.11.4 cumulative\_file\_delete - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

#### 2.9.11.5 Examples: Cumulate Files

```
[cumulate_using_cumulative_file]
include=local_source_host,local_target_host,jadeHistory
file_spec=^.*\.txt$
operation=copy
source_dir=R:/backup/sos/java/junittests/testdata/JADE/
target_dir=c:/temp/a
cumulate_files=true
cumulative_file_name=t.1
cumulative_file_separator=--- File: %{SourceFileName} ---
cumulative_file_delete=true
```

Profile "sample\_transfer"

## 2.10 Parameters for Pre- and Post-Processing

Parameter für Pre- und Post-Verarbeitung

## 2.10.1 Commands for the transfer

Commands for the transfer

### 2.10.1.1 pre\_transfer\_commmnds - pre\_transfer\_commmnds

Data-Type: SOSOptionCommandString

The default value for this parameter is .

### 2.10.1.2 post\_transfer\_commmnds - commands, which has to be executed after the transfer ended

Data-Type: SOSOptionCommandString

The default value for this parameter is .

### 2.10.1.3 Cumulate File Parameters

With the options described in this chapter one can be determined that the files from the resultlist not should be transferred separately, but are to be combined prior to transfer to another, further file (target file) consecutively.

The name of the file is set with the parameter `cumulative_filename` .

With the parameter `cumulative_file_separator` is set a text that will be inserted between the individual files in the target file.

The files can be deleted after transfer. This is determined with the parameter `cumulative_file_delete` .

#### 2.10.1.3.1 cumulate\_files - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.10.1.3.2 cumulative\_filename - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.



The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionOutFileName

The default value for this parameter is .

Use together with parameter: [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

#### 2.10.1.3.3 cumulative\_file\_separator - cumulate (all) files into a single file

This option causes files from the results list to be cumulated together into a single target file, one after the other, before being transferred. This is in contrast to the default transfer mode in which files are transferred individually.

The name of the cumulative file is set using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

#### 2.10.1.3.4 cumulative\_file\_delete - cumulate (all) files into a single file

This option determines whether the individual files from the results list should be individually transferred and then cumulated (i.e. packed together in a single file) at the transfer target.

The name of the cumulative file is specified using the [cumulative\\_filename](#) parameter.

The [cumulative\\_file\\_separator](#) parameter is used to specify a text that is placed in the target file, between the individual files, in order to be able to separate these files.

The individual files can be deleted after transfer if required. This is determined using the [cumulative\\_file\\_delete](#) or [remove\\_files](#) parameters.

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [cumulative\\_filename](#), [cumulative\\_file\\_separator](#), [cumulative\\_file\\_delete](#), [cumulate\\_files](#)

### 2.10.1.3.5 Examples: Cumulate Files

```
[cumulate_using_cumulative_file]
include=local_source_host,local_target_host,jadeHistory
file_spec=^.*\..txt$
operation=copy
source_dir=R:/backup/sos/java/junittests/testdata/JADE/
target_dir=c:/temp/a
cumulate_files=true
cumulative_file_name=t.1
cumulative_file_separator=--- File: %{SourceFileName} ---
cumulative_file_delete=true
```

Profile "sample\_transfer

### 2.10.1.4 Compressing Files

#### Compressing Files

#### 2.10.1.4.1 compress\_files

This parameter specifies whether or not the content of the source files should be compressed by using a zip-algorithm.

In case of sending files each files which has to be selected for sending will be compressed in a single zip file. The extension of the file-name is defined with the parameter compress\_filename\_extension.

A gzip-compatible compression is used, no further software components are required.

If compress\_files is used the parameter append\_files is not allowed.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.10.1.4.2 compressed\_file\_extension - Additional file-name extension for compressed files

This parameter specifies the file extension should target file compression be specified using the compress\_files parameter.

Data-Type: SOSOptionString

The default value for this parameter is **.gz**.

#### 2.10.1.4.3 decompress\_after\_transfer - Unpack Files after Transfer

Data-Type: SOSOptionString

The default value for this parameter is **false**.

## 2.10.2 Compressing Files

Commands for each object

### 2.10.2.1 Pre\_Command - Command to be executed before transfer

Data-Type: SOSOptionString

The default value for this parameter is .

### 2.10.2.2 Post\_Command - Command to be executed after transfer

Data-Type: SOSOptionString

The default value for this parameter is .

## 2.11 FTP only Parameters

Client Parameters

### 2.11.1 account - Additional accounting information

Data-Type: SOSOptionString

### 2.11.2 passive\_mode - passive\_mode

Passive mode for FTP is often used with firewalls.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.11.3 transfer\_mode - Type of Character-Encoding

Transfer mode is used for FTP exclusively and can be either `ascii` or `binary`.

Data-Type: SOSOptionTransferMode

The default value for this parameter is **binary**.

#### 2.11.4 alternative\_account - Additional accounting information

Alternative parameter for the primary parameter account .

Data-Type:

#### 2.11.5 alternative\_passive\_mode

Alternative parameter for the primary parameter passive\_mode .

Data-Type:

#### 2.11.6 alternative\_transfer\_mode

Alternative parameter for the primary parameter transfer\_mode.

Data-Type:

#### 2.11.7 check\_server\_features - Get a List of Server features

This command causes the FTP server to list all new FTP features that the server supports beyond those described in RFC 959. A typical example reply to the FEAT command might be a multi-line reply of the form:

```
C> FEAT
S> 211-Extensions supported
S> SIZE
S> MDTM
S> MLST size*;type*;perm*;create*;modify*;
S> LANG EN*
S> REST STREAM
S> TVFS
S> UTF8
S> 211 end
```

Data-Type: SOSOptionString

The default value for this parameter is **false**.

#### 2.11.8 protocol\_command\_listener - Activate the ftp-client logging

This Parameter can activate the logging of the apache commons ftp-client. All output will be written into the logging file.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

## 2.12 SSH only Parameters

### Client Parameters

#### 2.12.1 auth\_file

The value of this parameter specifies the path and name of a user's private key file used for authorisation on an SSH server. This parameter must be specified if the *publickey* authorization method has been specified in the *auth\_method* parameter. Should the private key file be secured by a password, then this must be specified using the *password* parameter.

Data-Type:

#### 2.12.2 auth\_method

Data-Type:

#### 2.12.3 strict\_hostkey\_checking - SSH HostKey checking

If this parameter is set to "yes", the application will not automatically add host keys to the `$HOME/.ssh/known_hosts` file, and will refuse to connect to hosts whose host key has changed.

This parameter forces the user to manually add all new hosts.

If this property is set to "no", the application will automatically add a new host key to the user known hosts files.

If this parameter is set to "ask", new host keys will be added to the user known host files only after the user has confirmed that is what they really want to do, and the application will refuse to connect to hosts whose host key has changed.

Setting this parameter to "yes" provides maximum protection against Trojan horse attacks.

See also <http://www.online-tutorials.net/security/secure-ssh-tutorial-part-1-host-key/tutorials-t-69-201.html> .

Data-Type: SOSOptionString

The default value for this parameter is **no**.

#### 2.12.4 SSH proxy parameter

##### 2.12.4.1 ssh\_proxy\_host

The value of this parameter is the host name or the IP address of a proxy that is used in order to establish a connection to the SSH server. The use of a proxy is optional.

Data-Type:

#### 2.12.4.2 ssh\_proxy\_port - Port number of proxy

This parameter specifies the port number of the proxy, should a proxy be used in order to establish a connection, e.g. to a SSH server.

Data-Type:

#### 2.12.4.3 ssh\_proxy\_user - User-ID for the Proxy-Server

The value of this parameter specifies the user account for authentication by the proxy server should a proxy be used in order to connect to the SSH server.

Data-Type:

#### 2.12.4.4 ssh\_proxy\_password - Password of Proxy-Server

This parameter specifies the password for the proxy server user account, should a proxy be used in order to connect to the SSH server.

Data-Type:

### 2.13 Params to control the Transfer History

Params to control the Transfer History

#### 2.13.1 Background Service Parameters

Client Parameters

##### 2.13.1.1 scheduler\_host

This parameter specifies the host name or IP address of a server for which Job Scheduler is operated for Managed File Transfer. The contents of an optional history file (see parameter history), is added to a central database by Job Scheduler. This parameter causes the transfer of the history entries for the current transfer by UDP to Job Scheduler. Should Job Scheduler not be accessible then no errors are reported, instead, the contents of the history will automaticall be processed later on.

Data-Type: SOSOptionHostName

The default value for this parameter is **localhost**.

##### 2.13.1.2 scheduler\_job\_chain - The name of a job chain

The name of a job chain for Managed File Transfer with Job Scheduler, see parameter scheduler\_host. The job chain accepts history entries and performs an import into a (central) database.

Data-Type: JSJobChain

The default value for this parameter is **scheduler\_sosftp\_history**.

### 2.13.1.3 scheduler\_port

The TCP-port for which a JobScheduler, see parameter scheduler\_host.

Data-Type: SOSOptionPortNumber

The default value for this parameter is **4444**.

### 2.13.1.4 send\_transfer\_history - send transfer history to the background service

send transfer history to the background service

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.13.1.5 scheduler\_transfer\_method - how to communicate with the BackgroundService

how to communicate with the BackgroundService. Possible values are "UDP" and "TCP".

Data-Type: SOSOptionString

The default value for this parameter is **UDP**.

## 2.13.2 Params to create a history file

Params to control the Transfer History

### 2.13.2.1 history - The name of the transfer history file

The value of this parameter specifies a history file. The history file is written in CSV format. The path and name of the history file is specified as value for this parameter. A history record is created for each file that has been transferred.

A history file contains the following columns:

**guid**

A unique identifier for the history entry. This identifier is used for checking of duplicate entries in combination with Job Scheduler for Managed File Transfer.

**mandator**

A character that denominates the mandator of a file transfer, see respective parameter.

**transfer\_timestamp**

The point in time when the transfer took place.

**pid**

The process id of the current process that executes the file transfer, see parameter [current\\_pid](#).

**ppid**

The process id of the parent of the process that executes the file transfer, see respective parameter.

**operation**

One of the operations *send* or *receive*, see respective parameter.

**localhost**

The name of the host on which this program is executed.

**localhost\_ip**

The IP address of the host on which this program is executed.

**local\_user**

The name of the user account for which this program is executed.

**Target (Remote)\_host**

The name of the host to/from which a transfer is executed, see parameter [host](#).

**Target (Remote)\_host\_ip**

The IP address of the host to/from which a transfer is executed, see parameter [host](#).

**Target (Remote)\_user**

The name of the user account for the Target (Remote) host, see parameter [user](#).

**protocol**

The protocol can be either *ftp*, *sftp* or *ftps*, see respective parameter.

**port**

The port on the Target (Remote) host, see respective parameter.

**local\_dir**

The local directory to/from which a file has been transferred, see respective parameter.

**Target (Remote)\_dir**

The Target (Remote) directory to/from which a file has been transferred, see respective parameter.

**local\_filename**

For *send* operations this is the original file name on the local host.

For *receive* operations this is the resulting file name on the local host optionally having applied [replacement](#) operations, see parameter [replacing](#).

**Target (Remote)\_filename**

For *send* operations this is the resulting file name on the Target (Remote) host optionally having applied [replacement](#) operations, see parameter [replacing](#).

For *receive* operations this is the original file name on the Target (Remote) Host.

**file\_size**

The size of the transferred file in bytes.

**md5**

The value of the MD5 hash that is created from the file that was transferred.

**status**

The status can be either *success* or *error*.



**last\_error\_message**

Should an error have occurred then the last error message will be given in this column.

**log\_filename**

The name of the log file, see respective parameter.

Data-Type: SOSOptionOutFileName

**2.13.2.2 history\_entries - additional entries for the transfer history**

additional entries for the transfer history

Data-Type: SOSOptionString

The default value for this parameter is .

**2.13.2.3 history\_repeat**

The parameter is used in order to synchronize parallel write access to the history file by multiple instances of this program. This parameter specifies the maximum number of repeat intervals when trying to write to the history file if the history file is locked due to parallel instances of this program.

Data-Type: SOSOptionInteger

The default value for this parameter is **3**.

**2.13.2.4 history\_repeat\_interval**

The parameter is used in order to synchronize parallel write access to the history file by multiple instances of this program. This parameter specifies the the interval in seconds of repeated trials to write to the history file if the history file is locked due to parallel instances of this program.

Data-Type: SOSOptionInteger

The default value for this parameter is **1**.

**2.13.2.5 mandator**

This parameter specifies the mandator for which a file transfer is effected. The mandator is added to an optional history file, see parameter history and has no technical relevance for the transfer.

Data-Type: SOSOptionString

The default value for this parameter is **SOS**.

**2.14 Polling Parameters**

Polling Parameters

### 2.14.1 poll\_interval - wait time interval between two poll-actions

This parameter specifies the time in seconds to wait, until a new try to poll for the files is made.

Data-Type: SOSOptionTime

The default value for this parameter is **60**.

### 2.14.2 poll\_minfiles - Minimum Number of Files expected

This parameter specifies the minimum number of files that have to be found during the polling period in order to cause the transfer to start. This parameter is effective only with the parameter poll\_timeout .

Data-Type: SOSOptionInteger

The default value for this parameter is **1**.

### 2.14.3 poll\_timeout - Max time to poll for files

This parameter specifies the time in minutes, how long a file is polled for. If the value of this parameter is zero a polling will not be done.

If a file becomes available within the time specified then it will be transferred, otherwise an error will be raised.

If a (or more) file(s) are received the polling will be stopped.

Data-Type: SOSOptionTime

The default value for this parameter is **0**.

### 2.14.4 poll\_keep\_connection - Keep connection while polling

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.14.5 waiting\_for\_late\_comers - Warten auf Nachzügler

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

### 2.14.6 poll\_error\_state - Name of next node in job-chain if polling failed

specify the name of an (error) state to which the order is set if the poll\_timeout has expired, without getting a file.

If no state is specified, an error will be raised.

Data-Type: SOSOptionString

The default value for this parameter is .

## 2.15 Virtual File System Parameters

### Client Parameters

#### 2.15.1 load\_Class\_Name - load\_Class\_Name: Name of the Data-Provider class

note\_en

```

; Test of parameter "loadClassName"
[ftp_receive_2_wilma]
include=wilma_host,wilma_user,do_sftp,BackgroundService,MailSettings
source_loadClassName=com.sos.VirtualFileSystem.SFTP.SOSVfsSftpJCraft
target_loadClassName=com.sos.VirtualFileSystem.SFTP.SOSVfsSftp
operation=receive
remote_dir=.
; file_spec=\.txt$
file_spec=.*
local_dir=C:/Users/kb/tmp/sosftp_inbound/

log_filename=${TEMP}/sosftphistory.log
    
```

Example: codeexample\_title\_en

codeexample\_title\_en

explanation\_en

Data-Type: SOSOptionJavaClassName

The default value for this parameter is .

Use together with parameter: [java\\_class\\_path](#)

#### 2.15.2 java\_class\_path - Java Class path

```

[ftp_receive_2_wilma]
include=wilma_host,wilma_user,do_sftp,BackgroundService,MailSettings
source_JavaClassPath=/home/scheulder/userlib/my.jar
source_loadClassName=com.sos.VirtualFileSystem.SFTP.SOSVfsSftpJCraft
operation=receive
remote_dir=.
; file_spec=\.txt$
file_spec=.*
local_dir=C:/Users/kb/tmp/sosftp_inbound/

log_filename=${TEMP}/sosftphistory.log
    
```

Example:

Data-Type: SOSOptionJavaClassName

The default value for this parameter is .

Use together with parameter: [load\\_Class\\_name](#)

### 2.15.3 buffer\_size - Max size of transferred data block

The maximum size of a data block is defined with this option.

Data-Type: SOSOptionInteger

The default value for this parameter is **32000**.

## 2.16 Global Parameters

List of parameters

Name	Title
<a href="#">buffer_size</a>	
<a href="#">log_filename</a>	
<a href="#">log4jpropertyfilename</a>	
<a href="#">verbose</a>	

Global params are usable for most of all applications.

### 2.16.1 buffer\_size - Max size of transferred data block

The maximum size of a data block is defined with this option.

Data-Type: SOSOptionInteger

The default value for this parameter is **32000**.

### 2.16.2 log\_filename - Name (and Pathname) of the logging-file

This parameter specifies the location of a file to which the log output will be written.

Should the file not exist then it will be created. If the file already exists then all log output will be appended.

Without specifying this parameter all log output will be written to stdout.

Data-Type: SOSOptionOutFileName

The default value for this parameter is **stdout, global**.

### 2.16.3 log4jPropertyFileName - Name (and Pathname) of a log4j configuration files

The configuration of the log4j framework is defined with this file. Generally there is a standard logging setting for applications.

If an application uses log4j and a configuration file has not been specified then a minimal configuration will be generated the first time the application is used. The configuration contains the following: Add an example standard configuration

Add general log4j text

Data-Type: SOSOptionInFileName

The default value for this parameter is **stdout, global**.

### 2.16.4 verbose - The granularity of (Debug-)Messages

Change granularity to granularity The *verbosity level* specifies the intensity of log entries. A value between 1 and 9 can be specified. Higher values cause more detailed information to be logged. Log output is written to stdout or to a file that has been specified with the parameter log\_filename.

Data-Type: SOSOptionInteger

The default value for this parameter is **10**.

## 2.17 Params of the JobScheduler JADE-Jobs

JobScheduler job specific Params

### 2.17.1 Connection parameters

#### 2.17.1.1 jump\_host

When using a jump\_host then files are first transferred to this host and then to the target system. Different protocols (FTP/SFTP) can be used for these transfer operations. Host or IP address of the jump\_host from which or to which files should be transferred in a first operation.

Data-Type:

#### 2.17.1.2 jump\_port

Port on the jump\_host by which files should be transferred. For FTP this is usually port 21, for SFTP this is usually port 22.

Data-Type:

The default value for this parameter is .

### 2.17.1.3 jump\_protocol

When using a jump\_host then files are first transferred to this host and then to the target system. Different protocols (FTP/SFTP) can be used for these transfer operations. This parameter expects ftp, sftp or ftps to be specified. If sftp is used, then the jump\_ssh\_\* parameters will be considered.

Data-Type:

The default value for this parameter is **sftp**.

### 2.17.1.4 proxy parameters

#### 2.17.1.4.1 jump\_proxy\_host

The value of this parameter is the host name or the IP address of a proxy used in order to establish a connection to the jump host. The use of a proxy is optional.

Data-Type:

#### 2.17.1.4.2 jump\_proxy\_port

This parameter specifies the port of a proxy that is used in order to establish a connection to the jump host, see parameter jump\_proxy\_host.

Data-Type:

## 2.17.2 Authentication parameters

### 2.17.2.1 jump\_user - User name for authentication with the jump host

User name for authentication with the jump\_host.

Data-Type:

### 2.17.2.2 jump\_password

Password for authentication with the jump\_host.

Data-Type:

### 2.17.2.3 proxy parameters

#### 2.17.2.3.1 jump\_proxy\_user

The value of this parameter specifies the user account for authentication by the proxy server should a proxy be used in order to connect to the jump host, see parameter jump\_proxy\_host.

Data-Type:

#### 2.17.2.3.2 jump\_proxy\_password

This parameter specifies the password for the proxy server user account, should a proxy be used in order to connect to the jump host, see parameter jump\_proxy\_host.

Data-Type:

#### 2.17.2.4 jump\_proxy\_password

This parameter specifies the password for the proxy server user account, should a proxy be used in order to connect to the jump host, see parameter jump\_proxy\_host.

Data-Type:

#### 2.17.2.5 jump\_proxy\_user

The value of this parameter specifies the user account for authentication by the proxy server should a proxy be used in order to connect to the jump host, see parameter jump\_proxy\_host.

Data-Type:

### 2.17.3 shell command parameters

#### 2.17.3.1 jump\_command - shell command string

This parameter specifies the command that is to be executed on the DMZ server to start the client.

Data-Type: SOSOptionCommandString

#### 2.17.3.2 jump\_command\_delimiter - Delimiter character/string for multiple shell commands

Data-Type: SOSOptionString

The default value for this parameter is %%.

This parameter is mandatory.

### 2.17.3.3 jump\_command\_script - Name of executable script file

This parameter `jump_command_script` can be used as an alternative to `jump_command`, `jump_command_delimiter` and `jump_command_script_file`. It contains script code which will be transferred to the remote host as a file and will then be executed there.

Data-Type: SOSOptionCommandScript

### 2.17.3.4 jump\_command\_script\_file

This parameter can be used as an alternative to `jump_command`, `jump_command_delimiter` and `jump_command_script`. It contains the name of a script file, which will be transferred to the remote host and will then be executed there.

Data-Type: SOSOptionCommandScriptFile

## 2.17.4 Secure Shell (SSH) parameters

### 2.17.4.1 jump\_ssh\_auth\_file

This parameter specifies the path and name of a user's private key file used for login to the SSH server of the `jump_host`. This parameter must be specified if the *publickey* authentication method has been specified in the `jump_ssh_auth_method` parameter. Should the private key file be secured by a password, then this password has to be specified using the `jump_password` parameter.

Data-Type:

### 2.17.4.2 jump\_ssh\_auth\_method

This parameter specifies the authentication method for the SSH server - the *publickey* and *password* methods are supported. When the *publickey* authentication method is used, then the path name of the private key file must be set in the `jump_ssh_auth_file` parameter. Should the private key file be secured by a passphrase then this passphrase has to be specified by the `jump_password` parameter. For the *password* authentication method the password for the user account has to be specified using the `jump_password` parameter. The authentication methods that are enabled depend on the SSH server configuration. Not all SSH servers are configured for *password* authentication.

Data-Type:

### 2.17.4.3 jump\_ignore\_error

Should the value *true* be specified, then execution errors caused by commands on the SSH server are ignored. Otherwise such execution errors will be reported.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.



#### 2.17.4.4 jump\_ignore\_signal

Should the value *true* be specified, then on Unix systems all signals will be ignored that terminate the execution of a command on the SSH server - if for example a command is terminated using *kill*. Note that by default errors will be reported for commands that are terminated by signals.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.17.4.5 jump\_ignore\_stderr

This job checks if any output to stderr has been created by a command that is being executed on the SSH server and reports this as an error. Should the value *true* be specified for this parameter, then output to stderr will not be reported as an error by the Job Scheduler.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.17.4.6 jump\_simulate\_shell

Should the value *true* be specified for this parameter, then a login to a shell is simulated to execute commands. Some scripts may cause problems if no shell is present.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

#### 2.17.4.7 jump\_simulate\_shell\_inactivity\_timeout

If no new characters are written to stdout or stderr after the given number of milliseconds, then it is assumed that the command has been carried out and that the shell is waiting for the next command.

Data-Type: SOSOptionInteger

#### 2.17.4.8 jump\_simulate\_shell\_login\_timeout

If no new characters are written to stdout or stderr after the given number of milliseconds, then it is assumed that the login has been carried out and that the shell is waiting for the next command.

Data-Type: SOSOptionInteger

#### 2.17.4.9 jump\_simulate\_shell\_prompt\_trigger

The expected command line prompt. Using this prompt the program tries to find out if commands may be entered or have been carried out. If no prompt can be configured, then timeout parameters have to be used in order to check if the shell is ready to accept commands.

Data-Type: SOSOptionString

## 2.18 eMail Notification Parameter

### List of parameters

Name	Title
<a href="#">mail_on_success</a>	
<a href="#">mail_on_error</a>	
<a href="#">mail_on_empty_files</a>	
<a href="#">ftp_file_notification_to</a>	
<a href="#">ftp_file_notification_cc</a>	
<a href="#">ftp_file_notification_bcc</a>	
<a href="#">ftp_file_notification_subject</a>	
<a href="#">ftp_file_notification_body</a>	

### 2.18.1 mail\_on\_success - Send a Mail on Success

```
mail_on_success=true
mail_on_success_to=kb@sos-berlin.com
mail_on_success_subject=Files transfered without errors
...
```

Data-Type: SOSOptionBoolean

### 2.18.2 mail\_on\_error - Send Mail in case of Error

```
mail_on_error=true
mail_on_error_to=kb@sos-berlin.com
mail_on_error_subject=Files transfered without errors
...
```

This parameter specifies the recipient of a notification mail, should files have been transferred.

Data-Type: SOSOptionBoolean

### 2.18.3 mail\_on\_empty\_files - eMail if empty files detected

```
mail_on_empty_files=true  
mail_on_empty_files_to=kb@sos-berlin.com  
mail_on_empty_files_subject=Files transfered without errors  
...
```

This parameter specifies the recipient of a notification mail, should files have been transferred.

Data-Type: SOSOptionBoolean

### 2.18.4 ftp\_file\_notification\_to - Recipient of a notification mail

This parameter specifies the recipient of a notification mail.

Data-Type: SOSOptionMailAdress

### 2.18.5 ftp\_file\_notification\_cc - eMail Notification: CC to ...

This parameter specifies the recipient of a carbon copy notification mail, should files have been transferred.

Data-Type: SOSOptioneMailAdress

### 2.18.6 ftp\_file\_notification\_bcc - Bcc eMail Recipient

This parameter specifies the recipient(s) of a blind carbon copy notification e-mail , should files have been transferred

Data-Type: SOSOptioneMailAdress

### 2.18.7 ftp\_file\_notification\_subject - Subject of an eMail Notification

This parameter specifies the subject of a mail notification, should files have been transferred.

Data-Type: SOSOptionString

### 2.18.8 ftp\_file\_notification\_body - eMail Body

This parameter specifies the body of a mail notification, should files have been transferred.

The list of files transferred will be added to the mail body.

Data-Type: SOSOptionString

### 3 JobScheduler JADE-Job Parameters

JobScheduler job specific parameters

#### 3.1 File Orders

##### 3.1.1 File Order Creation

List of parameters

Name	Title
<a href="#">create_order</a>	
<a href="#">create_orders_for_all_files</a>	
<a href="#">order_jobchain_name</a>	
<a href="#">next_state</a>	
<a href="#">merge_order_parameter</a>	

###### 3.1.1.1 create\_order - Activate file-order creation

With this parameter it is possible to specify, that for all filenames in the resultlist or for the first file only (see [create\\_orders\\_for\\_all\\_files](#) ) a file-order has to be created and launched.

For more information on "file-orders" see [Directory Monitoring with File Orders](#).

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [create\\_orders\\_for\\_all\\_files](#), [order\\_jobchain\\_name](#), [next\\_state](#), [merge\\_order\\_parameter](#)

###### 3.1.1.2 create\_orders\_for\_all\_files - Create a file-order for every file in the result-list

If this parameter is set to "true", for each file which which is in the result set, a file-order is created and started.

This parameter is in effect only if the parameter [create\\_orders](#) has the value "true".

```
create_orders=true
create_orders_for_all_files=true
```

Example: example 1: create a file-order

Whith this example for all files in the result list a file-order will be created.

Valid values: true, 1, on, yes, y, ja, j and false, 0, off, no, n, nein.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [create\\_order](#), [order\\_jobchain\\_name](#), [next\\_state](#)

### 3.1.1.3 order\_jobchain\_name - The name of the jobchain which belongs to the order

The name of the jobchain which has to be launched by the order is the value of this parameter.

One must take into account, that the name of the jobchain must contain a subfolder structure if the jobchain is not in the folder "live". An example: the jobchain "Test" is located in "live/sample/FileOperations/". The value which has to be specified is then "/sample/FileOperations/Test".

Data-Type: SOSOptionString

The default value for this parameter is .

Use together with parameter: [create\\_order](#), [next\\_state](#)

### 3.1.1.4 next\_state - The first node to execute in a jobchain

The name of the node of a jobchain, with which the execution of the chain must be started, is the value of this parameter.

Data-Type: SOSOptionJobChainNode

The default value for this parameter is .

Use together with parameter: [create\\_order](#), [order\\_jobchain\\_name](#)

### 3.1.1.5 merge\_order\_parameter - merge actual order parameter into new created order

This parameter specifies, that the order, which has to be created, will be extended by the parameters of the actual order.

Data-Type: SOSOptionBoolean

The default value for this parameter is **false**.

Use together with parameter: [create\\_order](#)

## 3.1.2 File Order Parameter

List of parameters

Name	Title
<a href="#">scheduler_file_path</a>	
<a href="#">scheduler_file_parent</a>	

Name	Title
<a href="#">scheduler_file_name</a>	

### 3.1.2.1 scheduler\_file\_path - file to process for a file-order

Using [Directory Monitoring with File Orders](#) the job parameter [scheduler\\_file\\_path](#) contains automatically the path of the file that triggered the order.

Data-Type: SOSOptionFileName

The default value for this parameter is **empty**.

### 3.1.2.2 scheduler\_file\_parent - pathname of the file to process for a file-order

This paramters contains the pathname (parent) of the file which is currently processed.

Data-Type: SOSOptionFileName

The default value for this parameter is **empty**.

### 3.1.2.3 scheduler\_file\_name - Name of the file to process for a file-order

The name, without foldername, of the file which is currently processed.

Data-Type: SOSOptionFileName

The default value for this parameter is **empty**.

## 3.2 Result Set

List of parameters

Name	Title
<a href="#">on_empty_result_set</a>	
<a href="#">expected_size_of_result_set</a>	
<a href="#">raise_error_if_result_set_is</a>	
<a href="#">result_list_file</a>	

### 3.2.1 on\_empty\_result\_set - Set next node on empty result set

The next Node (Step, Job) to execute in a JobChain can be set with this parameter. The value of the parameter is a (valid) node-name of the current JobChain. In case of an empty result-set, e.g. due to non existent files, the current job will end without an errors and the JobChain will continue with the name of the node which is given as the value of this parameter.

Data-Type: SOSOptionJobChainNode

The default value for this parameter is **empty**.

### 3.2.2 `expected_size_of_result_set` - number of expected hits in result-list

Data-Type: SOSOptionInteger

The default value for this parameter is **0**.

Use together with parameter: [raise\\_error\\_if\\_result\\_set\\_is](#)

### 3.2.3 `raise_error_if_result_set_is` - raise error on expected size of result-set

With this parameter it is possible to raise an error if the quantity of hits of the result-list is according to the value of this parameter.

An example:

Assuming, that the parameter "`raise_error_if_result_set_is=ne`" is defined and the parameter "`expected_size_of_result_set=1`" is specified as well. If the number of hits is not equal to "1" an error will raised.

Data-Type: SOSOptionRelOp

The default value for this parameter is **.**

Use together with parameter: [expected\\_size\\_of\\_result\\_set](#)

### 3.2.4 `result_list_file` - Name of the result-list file

If the value of this parameter specifies a valid filename the result-list will be written to this file.

Data-Type: SOSOptionFileName

The default value for this parameter is **empty**.

## 3.3 Return Parameter

Return Parameter

### 3.3.1 Parameters

List of parameters

Name	Title
<a href="#">scheduler_sosfileoperations_resultset</a>	

Name	Title
<a href="#">scheduler_sosfileoperations_resultsetsize</a>	
<a href="#">scheduler_sosfileoperations_file_count</a>	

3.3.1.1 scheduler\_sosfileoperations\_resultset - The result of the operation as a list of items

Data-Type: SOSOptionString

The default value for this parameter is **empty**.

Use together with parameter: [scheduler\\_sosfileoperations\\_Resultsetsize](#), [scheduler\\_sosfileoperations\\_file\\_count](#)

3.3.1.2 scheduler\_sosfileoperations\_resultsetsize - The amount of hits in the result set of the operation

Data-Type: SOSOptionInteger

The default value for this parameter is **empty**.

Use together with parameter: [scheduler\\_sosfileoperations\\_ResultSet](#), [scheduler\\_sosfileoperations\\_FileCount](#)

3.3.1.3 scheduler\_sosfileoperations\_file\_count - Return the size of the result set after a file operation

Data-Type: SOSOptionInteger

The default value for this parameter is **0**.

Use together with parameter: [scheduler\\_sosfileoperations\\_Resultsetsize](#), [scheduler\\_sosfileoperations\\_ResultSet](#)



# Index

- A**
  - atomic\_prefix, parameter 16, 17, 31, 33
  - atomic\_suffix, parameter 16, 17, 31, 33
  - auth\_file, parameter 13, 45
  - auth\_method, parameter 13, 45
- B**
  - buffer\_size, param 52, 52
- C**
  - check\_server\_features 44
  - concurrent\_transfer 17
  - Configuration, log4j 53
  - cumulate\_files, param 19, 38, 40
  - cumulative\_file\_delete, param 20, 39, 41
  - cumulative\_file\_delete, parameter 19, 38, 40
  - cumulative\_file\_separator, param 19, 38, 41
  - cumulative\_file\_separator, parameter 19, 38, 40
  - cumulative\_filename, param 19, 38, 40
  - cumulative\_filename, parameter 19, 38, 40
- D**
  - decompress\_after\_transfer 18, 42
- E**
  - eMail, Parameter 58
  - Encoding 24
  - ErrorOnNoDataFound, parameter 25
- F**
  - FEAT 44
  - file\_list\_name, parameter 15
  - file\_path, parameter 14
  - file\_spec, parameter 15, 25
  - file, parameter 25
  - FileNamePatternRegExp, parameter 15
  - force\_files, parameter 25, 32
- H**
  - history\_entries 49
- J**
  - jump\_command\_script 56
- K**
  - Keyboard-Interactive 0, 0
  - Konfiguration, log4j 0
- L**
  - log\_filename, param 52
  - log4j, Configuration 53
  - log4j, Konfiguration 0
  - log4jPropertyFileName, param 53
- M**
  - mail\_on\_empty\_files, Parameter 59
  - mail\_on\_error, Parameter 58, 58
  - MailOnEmptyFiles\_, Prefix 58, 59
  - MailOnError\_, Prefix 58, 58
  - MailOnSuccess\_, Prefix 58, 58
  - max\_concurrent\_transfers 18
- O**
  - overwrite\_files, parameter 32
- P**
  - Parallel Transfer 17
  - param, buffer\_size 52, 52
  - param, cumulate\_files 19, 38, 40
  - param, cumulative\_file\_delete 20, 39, 41
  - param, cumulative\_file\_separator 19, 38, 41
  - param, cumulative\_filename 19, 38, 40
  - param, log\_filename 52
  - param, log4jPropertyFileName 53
  - parameter, atomic\_prefix 16, 17, 31, 33
  - parameter, atomic\_suffix 16, 17, 31, 33
  - parameter, auth\_file 13, 45
  - parameter, auth\_method 13, 45
  - parameter, cumulative\_file\_delete 19, 38, 40
  - parameter, cumulative\_file\_separator 19, 38, 40
  - parameter, cumulative\_filename 19, 38, 40
  - parameter, ErrorOnNoDataFound 25
  - parameter, file 25
  - parameter, file\_list\_name 15
  - parameter, file\_path 14
  - parameter, file\_spec 15, 25
  - parameter, FileNamePatternRegExp 15
  - parameter, force\_files 25, 32
  - parameter, overwrite\_files 32
  - parameter, Post\_Command 43
  - parameter, post\_transfer\_commands 40
  - parameter, Pre\_Command 43
  - parameter, profile 10
  - parameter, ProfileName 10
  - parameter, RecurseSubFolders 25
  - parameter, recursive 25
  - parameter, remote\_dir 33
  - parameter, remove\_files 19, 38, 40
  - parameter, replacement 20, 22, 26, 28, 34, 35
  - parameter, SectionName 10
  - parameter, settings 10
  - parameter, ssh\_proxy\_host 13, 45
  - parameter, ssh\_proxy\_password 14, 46
  - parameter, ssh\_proxy\_port 14, 46
  - parameter, ssh\_proxy\_user 14, 46
  - parameter, StrictHostkeyChecking 13, 33, 45
  - parameter, transactional 15, 31, 31

parameter, transfer\_mode 43  
parameter, waiting\_for\_late\_comers 50  
password, auth\_method 13, 45  
Post\_Command, parameter 43  
post\_transfer\_commmands, parameter 40  
Pre\_Command, parameter 43  
pre\_transfer\_commmands, parameter 40  
profile, parameter 10  
ProfileName, parameter 10  
publickey, auth\_method 13, 45

## R

RecurseSubFolders, parameter 25  
recursive, parameter 25  
remote\_dir, parameter 33  
remove\_files, parameter 19, 38, 40  
replacement, parameter 20, 22, 26, 28, 34, 35

## S

scheduler\_transfer\_method 47  
SectionName, parameter 10  
send\_transfer\_history 47  
settings, parameter 10  
settings, pre\_transfer\_commmands 40  
ssh\_proxy\_host, parameter 13, 45  
ssh\_proxy\_password, parameter 14, 46  
ssh\_proxy\_port, parameter 14, 46  
ssh\_proxy\_user, parameter 14, 46  
SSH, auth\_method 13, 45  
SSH, ssh\_proxy\_host 13, 45  
SSH, ssh\_proxy\_password 14, 46  
SSH, ssh\_proxy\_port 14, 46  
SSH, ssh\_proxy\_user 14, 46  
StrictHostkeyChecking, parameter 13, 33, 45

## T

transactional, parameter 15, 31, 31  
transfer\_mode, parameter 43

## W

waiting\_for\_late\_comers 50