StreamServe™ Persuasion™

StreamServe Persuasion SP5 Startup arguments

Reference Guide

Rev D

StreamServe Persuasion SP5 Startup arguments Reference Guide
Rev D

Contents

Startup argument reference

The StreamServer needs a number of startup arguments. It finds the arguments in the startup argument file (`*.arg`). Startup arguments are automatically defined and added to the startup argument file when you export the Project from the Design Center. You can also specify arguments manually by selecting **Platform > Configure Export**.

A

-a

Syntax `-a <file_name>`

Description Specifies an argument file for the StreamServer to read and process.

Comment Command line argument if the server runs stand-alone. This argument cannot be used if you start the StreamServer from Control Center.

Example `-a start.arg`

-alternativebarcode

Syntax `-alternativebarcode`

Description Specifies to use Adobe style barcodes which are used from SP2 and onwards. This argument is used by default if the Project has been upgraded from an SP2, SP3, or SP4 Project.

Comment -

Example `-alternativebarcode`

-alternativequeuing

Syntax `-alternativequeuing`

Description Combines scheduled spooling with notified spooling in a shared queue environment. That is, when a StreamServer application queues data, the same application is notified and immediately tries to retrieve and process the data.

The argument enables the following optimizations:

- Better utilization of the stream cache in a shared queue environment.
- Reduced number of database readings from the runtime repository.
- Enhanced performance.

Comment You must enable schedule spooling in the Manage Queues dialog box in Design Center. The StreamServer application polls the queue at the configured spooling interval. Once the application receives input data, the configured interval is no longer considered for as long as there is data in the queue. When all the data is de-queued the server will start polling at the defined interval again. Since the application de-queues data directly it is not possible to use the argument together with scheduled spooling at specific intervals of for a limited time.

The argument has no effect on the service queue (used in Projects that include service-enabled Messages or Service Request input connectors).

Example `-alternativequeuing`

-args

Syntax `-args <file_name>`

Description Specifies an argument file for the StreamServer to read and process (see `-a`).

Comment Command line argument if the server runs stand-alone. This argument cannot be used if you start the StreamServer from Control Center.

This argument is identical to the `-a` argument.

Example `-args start.arg`

-asynchronqueue

Syntax `-asynchronqueue <maxsize>`

Description Set the maximum number of concurrent asynchronous requests allowed for a processing job. Default is 10 requests.

Comment Each request handles writing of an output job to the output queue. The asynchronous queue uses the core IO dispatch thread pool to perform the asynchronous requests (configured in threadmanager.xml).

You can reduce the number of asynchronous requests if the processing job is creating too much load on the database server and the output queue. The number of asynchronous requests could also be reduced if the server simultaneously handles several highly loaded output queues.

Note: Asynchronous queuing can only be applied to output queues.

Example `-asynchronqueue 5`

D

-demo

Syntax `-demo`

Description Runs the StreamServer in demo mode, which does not require any license.

Comment In the output, the text “demo” is randomly included.

Example `-demo`

-disableasynchronqueuing

Syntax `-disableasynchronqueuing`

Description Disable asynchronous queuing for all output queues.

Comment By disabling asynchronous queuing, the processing job will write all output jobs in sequential order to the output queue. The job processing thread will take all IO waits preventing it from formatting data. Asynchronous queuing will increase performance for batch jobs 1:M. If the server mostly runs 1:1 jobs, you should consider disabling asynchronous queuing.

Note: Asynchronous queuing can only be applied to output queues.

-disablestreamcache

Syntax `-disablestreamcache`

Description Disable the stream cache for all queues, and read all data from the runtime repository.

Comment

The stream cache is by default used by all queues. This argument is global, and is applied to all queues used by the StreamServe application.

The stream cache is an optimization used to reduce database readings when the same server queues and processes queue items. In a situation where several servers are spooling queue items from the same queue, it is not guaranteed that the same server that placed an item in the queue is the one picking it up for processing. If a different server picks up the queue item for processing, it will introduce a cache miss, forcing the processing server to read back the data from the database. If this occurs frequently, it could be beneficial to disable the stream cache. The stream cache can also be disabled in order to reduce used memory and resources.

-dumpvars

Syntax

`-dumpvars <context>`

Description

Dumps variables and their values to a text file. Only variables assigned to a value at or before the specified context are written to this file.

The context is represented by a hex value listed in the table below. To dump the variables at all contexts, use hex value `0xFFFF`.

A file with the dumped variables and their values is created in the deployed Project's working directory.

The file name is `strs_dump_vars<job_ID>.txt`.

Comment

The Preproc phase (0x01) can only be used in combination with other contexts.

Example

```
//Enable dumping of variables assigned Before Process,
execution and Preproc phases
-dumpvars 0x05 // (0x04 and 0x01)
//Enable dumping of variables assigned Before Process,
Retrieved/Collect and execution phase
-dumpvars 0x06 // (0x04 and 0x02)
//Enable dumping of variables assigned After Event, execution
and Preproc phases
-dumpvars 0x21 // (0x20 and 0x01)
```

Context	
Preproc	0x01
Retrieved / Collect	0x02

Context	
Before Process	0x04
After Process	0x08
Before Event	0x10
After Event	0x20
Before Job	0x40
After Job	0x80

E

-education

Syntax `-education`

Description Runs an unlicensed version of the StreamServer in education mode for fourteen days. After the fourteen-day period has ended, you must either use a license file or run it in demo-mode (`-demo`).

Example `-education`

-evaluation

Syntax `-evaluation`

Description Runs an unlicensed version of the StreamServer in evaluation mode for fourteen days. After the fourteen-day period has ended, you must either use a license file or run it in demo mode (`-demo`).

Note: Only applicable when the StreamServer runs in a Windows environment.

Example `-evaluation`

G

-grb

Syntax `-grb <path>`

Description Only applicable if you record sample files.

Specifies where sample files are saved when you run the StreamServer with the startup arguments `-rec` or `-reconly`.

Example `-grb C:\StreamServe\Server\grb`

-grbcodepage

Syntax `-grbcodepage <code_page>`

Where `<code_page>` is the name of the new code page.

Description Only applicable if you record sample files.

Converts the code page of a recorded sample file to the code page included in the argument.

Comment Use this startup argument when you record sample files, where the code page in the sample file is not supported by StreamServe. You can then use the recorded sample file when you configure the StreamServe Project.

For more information about recording sample files, see the *PageIN* documentation.

Example `-grbcodepage cp866_DOSCyrillicRussian`

I

-ignorejobdefs

Syntax `-ignorejobdefs <jobdef1> [:<jobdef2>:<jobdef3>,...]`

Description Ignores all settings done for a job definition. For example, runtime connector and archiver settings. By ignoring job definition settings, variables defined in a specific job keep their values across different Messages in different job definitions.

Comment Use this startup argument if you use an upgraded Project from 3.0.1 or earlier where one or more Messages did not belong to a job definition. By ignoring job definition settings this way, you emulate the behavior of the 3.0.1 (or older) version where variable values were kept from previous Messages in the Messages not belonging to any job definition.

If you specify to *-includejobdefs* for the same job definition, the ignore setting overrules the include setting.

Note: This startup argument is deprecated and it may be removed in future releases of StreamServe software without any prior notice given.

Example `-ignorejobdefs jd1:jd2:jd4`

-includejobdefs

Syntax `-includejobdefs <jobdef1> [:<jobdef2>:<jobdef3>,...]`

Description Does the opposite from *-ignorejobdefs* by including all settings done for a job definition. This means also that all other job definitions in the job will be ignored.

Comment If you specify to *-ignorejobdefs* for the same job definition, the ignore setting overrules the include setting.

Note: This startup argument is deprecated and it may be removed in future releases of StreamServe software without any prior notice given.

Example `-includejobdefs jd3:jd5`

J

-java-options

Syntax `-java-options <property>`

Description Specifies the initial naming factory property for JNDI (Java Naming and Directory Interface).

Example
```
-java-options
Djava.naming.factory.initial=com.sun.jndi.fscontext.RefFSCon
textFactory
```

-java-user-class-path

Syntax `-java-user-class-path <path> <path>`

Description Specifies additional paths, libraries and JAR-files from which java classes can be loaded. This is the same as the java -classpath argument.

Example Windows:

```
-java-user-class-path c:\jndi.jar; c:\myjavaclasses
```

Unix:

```
-java-user-class-path /usr/local/app/jndi.ar:/opt/
streamserve/myjavaclasses
```

L

-langfile

Syntax

`-langfile <file_name.sls> [,<lang_code>,<lang_code>,...]`

- <*filename*.sls> specifies the name of the language file (`*.sls`)
- <*lang_code*> specify the language codes you want to use.

Description

Used with `*.sls` files. Specifies the name of the language file, and the language files within that file, to use.

Only applicable if you use StreamServe Language Sets files in the Project.

Only applicable for PageOUT.

Enables the StreamServer to dynamically change the language used in a PageOUT Process.

Comment

If you specify language codes, the StreamServer will ignore any language code *not* specified here. However, if you do not specify any language codes at all, the StreamServer will read all language codes specified in the StreamServe Language Sets file.

Example

```
-langfile language.sls,eng,swe
```

-ldapsslcertdb

Syntax

`-ldapsslcertdb <file>`

Where `<file>` is the path to the `cert7.db` certificate database.

Description

Enables the use of the `LdapConnectSSL` script function to authenticate the connection to the LDAP server.

Comment

This argument is required if you are setting up SSL communication between the StreamServer and the Sun(IPlanet)Directory Server(LDAP Server). Other LDAP servers are not supported. Follow the instructions in the manual for setting up SSL.

Example

```
-ldapsslcertdb cert7.db
```

-ldapSslKeyDb

Syntax `-ldapSslKeyDb <file>`

Where `<file>` is the path to the `key3.db` key database.

Description Enables the use of the `LdapConnectSSLCCA` script function to authenticate the connection to the LDAP server.

Comment You must use this argument together with the `-ldapsslcertdb` argument.

This argument is required if you are setting up SSL communication between the StreamServer and the Sun(IPlanet)Directory Server(LDAP Server). Other LDAP servers are not supported. Follow the instructions in the manual for setting up SSL.

Example
```
-ldapsslcertdb cert7.db
-ldapSslKeyDb key3.db
```

-licfile

Syntax `-licfile <filename>`

Description Specifies the license file.

Comment Can only be used as an argument when you start the StreamServer from command line, that is it can not be used in the startup argument file (`*.arg`).

Example
```
-licfile c:\streamserve\lic\strs.lic
```

-localpersistpath

Syntax `-localpersistpath <path>`

Description All `LOCAL` mode repositories and files used locally to generate unique IDs are stored under `<exportdir>\data\data` by default.You can move the directory to a different location, and use this startup argument to specify the new path to the directory.

Comment If any of the following directories have been set up in an earlier StreamServe installation:

- `<exportdir>\data\jr`
- `<exportdir>\data\transactions`

you must stop the StreamServer, and move these directories to the new location, before changing the repository path.

Example `-localpersistpath C:\localdata\repositories`

-logcp

Syntax `-logcp <code_page>`

Description Specifies a code page for the StreamServer log.

Comments

- Applies to StreamServers run from the command line.
- If the characters displayed in the log conform to Latin 1 you do not have to specify this argument.

Example `-logcp cp862_DOSGreek`

-logfilecp

Syntax `-logfilecp <code_page>`

Description Specifies a code page for the StreamServer log file.

Comments

- Applies to StreamServers run from Control Center.
- If the characters displayed in the log conform to Latin 1 you do not have to specify this argument.
- To enable Control Center to display “non-Latin 1” characters correctly in the log, you must specify UTF-8 as code page.

Example `-logfilecp UTF-8`

-lotusnotes

Syntax `-lotusnotes`

Description Only applicable if you use a Lotus Notes output connector.

Enables the use of Lotus Notes output connectors.

Example `-lotusnotes`

-lxfcachedynamic

Syntax `-lxfcachedynamic`

Description Turns on caching of dynamic overlays. Dynamic overlays are not cached by default because they decrease the performance of static overlays during larger jobs.

Comments When enabled, dynamic overlays stay in the cache between the preprocess and runtime phases and are not removed until after runtime is finished.

Example `-lxfcachedynamic`

-lxfcachesize

Syntax `-lxfcachesize <n>` where `<n>` must be an integer greater than zero.

Description Controls the number of cache items (LXF documents) that can be stored in the server.

Comments The cache will always try to cache static overlays. Static overlays stay in the cache as long as possible, and are only discarded when the server is shut down.

Dynamic overlays (created in PreformatIN) are not cached by default because they decrease the performance of static overlays during larger jobs. To cache dynamic overlays, see `lxfcachedynamic`.

Example `-lxfcachesize 20`

M

-maxinfiles

Syntax `-maxinfiles <n>`

Description Number of files that are scanned when directory scan is used.

The server exits when the amount of files are scanned.

Comment For test only.

Example `-maxinfiles 5`

-maxsortnodes

Syntax `-maxsortnodes <n>`

Description Assigns a maximum number of sorting nodes.

Comment Only applicable with sorting.

An internal list with sorting keys consumes internal memory. You can limit the size of the list to save time (less nodes are allocated).

-mbytefile

Syntax `-mbytefile <path>`

Where <*path*> is the full path and file name to the required multibyte file.

Description A unicode startup argument. Used if the multibyte file `ssmbyte.dat` is not located in any of its ordinary directories, or if you want to use another file.

Comment The StreamServe installation includes a default file, `ssmbyte.dat`, containing conversions between Unicode and multibyte code pages.

The StreamServer will try to locate `ssmbyte.dat` according to the operating system platform you are using. However, if the file is located elsewhere, or if you want to use another file, you can specify a different location by using the startup argument: `-mbytefile`

Example `-mbytefile <path>`

N

-norecgrb

Syntax `-norecgrb`

Description This argument is used together with the `-rec` argument.

The `-norecgrb` argument instructs the StreamServer not to generate separate sample files for each input page. See the `-rec` argument for more information.

Example `-norecgrb`

-nstack

Syntax `-nstack <n>`

Description Specifies the size of the script expression size stack. Default is 50.

Comment Rarely used. You can receive an “overflow” by using extremely large mathematical expressions.

O

-odbctimeout

Syntax `-odbctTimeOut <n>`

Description Specifies the time-out value that the StreamServer uses for connecting to data sources. The value is in seconds.

Comment If no value is specified, the default value, 30 seconds, is used.

Note: After a connection has been established, this argument is not applicable.

Example `-odbctimeout 60`

-optalias

Syntax `-optalias <file_name>`

Description Overrides the default overlay alias file name (`optalias`) with the specified file name.

Comment `optalias` is a default alias file name.

`optalias` is a lookup table. This could be another way of loading a lookup table.

Example `-optalias new_alias.txt`

-overlayfirstonpage

Syntax `-overlayfirstonpage`

Description Used with PCL and LIPS drivers to print the overlay before the processed output, for example if the PCL output contains white text printed over overlay objects.

Comment Overlays are sent to the driver before the processed output, except for the PCL and LIPS drivers. If you use the PCL or LIPS driver and you want the overlay to be sent to the driver before the overlay, you must use the `overlayfirstonpage` argument.

Example `-overlayfirstonpage`

P

-parse

Syntax `-parse`

Description Reads syntax to see if it is correct and then exits.

Comment Use this argument for syntax testing before runtime. For example to test `*.dux` files.

-pcl2pdfarg

Syntax `-pclpdfarg -ru:1`

Description Only applicable to PDF (PCL Convert) output.

If you have specified to download a soft font file when data is sent to a output connector, and have selected PDF (PCL Convert) as the device, you must add this argument.

Example `-pclpdfarg -ru:1`

-pid

Syntax `-pid <filename>`

Description Enables the StreamServer to write its PID in the specified file at start-up. The PID can be used later for termination.

Example `-pid <filename>`

-preloadmorefontdata

Syntax `-preloadmorefontdata`

Description Improves performance by preloading all font data required for text layout calculations in the XFA Processor. This will generally reduce the need for drivers to load additional font data which also can improve performance. However, an increase in startup time and runtime memory consumption can be expected.

Example `-preloadmorefontdata`

-prn

Syntax `-prn <path>`

Description Only applicable if you use PRN overlays.

Specifies the path for overlay files.

Comment

- The path is specified in relation to the *Application root* directory.
- If you insert the `-prn` argument before the <*file_name.dua*> argument, this argument overrides the directory specified in the StreamServe configuration.

Example `-prn files`

-prnalias

Syntax `-prnalias <file_name>`

Description Overrides the default overlay alias file name (`prnalias`) with the specified file name.

Comment `prnalias` is a default alias file name.

`prnalias` is a lookup table. This could be another way of loading a lookup table.

Example `-prnalias new_alias.txt`

Q

-quealias

Syntax `-quealias <file_name>`

Description Overrides the default connector alias file name (`quealias`) with the specified file name.

Example `-quealias new_alias.txt`

R

-rec

Syntax `-rec`

Description Instructs the StreamServer to record the input data and create sample files. In addition to creating the sample files, the StreamServer also creates ordinary output data via a PageIN Event and an appropriate Process. If you only want to create a sample file, and no output data, you must use the `-reconly` argument instead of the `-rec` argument.

The sample files are created in the directory specified by the `-grb` argument. The following sample files are created:

- `<event name><sequence number>.grb`. Each page recorded generates a separate sample file. You can use the `-norecgrb` argument together with `-rec` if you do not want to generate these sample files.
- `allpages.<input connector name>`. Each page recorded is appended to this sample file.

Comment The number of columns and lines that are recorded and included on one page in the `<event name><sequence number>.grb` file, is based on the page size in the PageIN configuration:

- Columns outside the specified page size are discarded.
- Lines outside the specified page size are included in the following `<event name><sequence number>.grb` file.

The `allpages.<input connector name>` sample file is not affected by the page size specified in the PageIN configuration. All columns and lines are recorded and included in the file.

Example `-rec`

-reconly

Syntax `-reconly <columns>,<lines>`

Description Instructs the StreamServer to record the input data and create a sample file. If you also want to create output data you must use the `-rec` argument instead of `-reconly`.

All pages recorded are appended to the sample file `allpages.<input connector name>`. This file is created in the directory specified by the `-grb` argument.

Comment You must specify values for `<columns>,<lines>`. These values are only for backward compatibility and are ignored by the StreamServer.

You can also specify record-only mode in the Platform configuration.

For more information about how to use the `-reconly` startup argument, see the *PageIN* documentation.

Example
```
-reconly 140,100
```

-reducenotifications

Syntax
```
-reducenotifications
```

Description Instructs the server to only generate notifications with a type requested by an external observer or a Status Messenger input connector.

Comment When running large input jobs it is more efficient to use the `reducenotifications` than to store all notifications and let the observer choose among them. The methods may be combined.

Example
```
-reducenotifications
```

-rmlog

Syntax
```
-rmlog <file>
```

Description Ensures that the logfile is deleted and recreated at startup.

Comment The `-rmlog` argument does not work if you run the StreamServer as a service.

Example
```
-rmlog ./logs/strs.log
```

S

-shareddatapath

Syntax `-shareddatapath`

Description Specifies the installation catalogue with the configuration files.

Comment The "config" in the server directory is default in Unix. This option is used to specify another directory path for configuration files.

Example `-shareddatapath <config directory>`

-sortdef

Syntax `-sortdef <filename>`

Description If you use the **sort** script function to sort processes, you must include an ASCII file, usually named `sortdef` in the startup argument file. The `sortdef` argument contains sort definitions.

Example `-sortdef sortdef.txt`

-sprog

Syntax `-sprog <n>`

Where `<n>` specifies a number greater than 500.

Description Allocates memory for scripts, for example the total number of bytes in the memory area. The default size is 10,000.

Comment Only use this argument if you need to allocate more than 10,000 bytes.

Example `-sprog 25000`

-statusevent

Syntax `-statusevent 0`

Description .Disables processing of ready status events. This means that no “Job completed” messages are added to the log and no status messenger events are consumed by the server instance.

Note: In the Job status configuration dialog. you must set **Report status** to **When delivered from the output queue**

-statusreporter

Syntax `-statusreporter`

Description Tracking and status reporting in a shared queue environment requires that one StreamServer is defined as the status reporting server. You use this startup argument to define a StreamServer as the status reporting server.

-stdin

Syntax `-stdin <file_name>`

Description Assigns a file to the StdIn file descriptor.

Comment This argument is only valid when using the StdIn input connector.

Equivalent with redirection in `CommunicationServer <file_name>`

-streamcache

Syntax `-streamcache <maxsize>`

Description Set the maximum number of cached streams per queue. Default is 200.

Comment This argument is global, and is applied to all queues used by the StreamServer application.

The stream cache is used as an optimization between the input queue and the job, and between the job and the output queue. You can reduce the number of cached streams per queue if the server runs out of temporary stream objects. Reducing the number of cached streams per queue will also reduce the memory and resource consumption.

Example `-streamcache 100`

-sync

Syntax `-sync`

Description Synchronizes output jobs with the input processing. The status of the output job is propagated back to the processing job. This argument can be necessary to get correct status returned by the `GetJobStatus()` script function.

Comment This argument can reduce performance of the server.

The jobs derived from one input connector are always processed in sequence, even if several threads are configured for the input queue. For most input connectors, this means that the connector runs single-threaded (even though several input connectors still run in parallel). Exceptions are the HTTP, HTTPS, and Service Request input connectors, which have their own thread pools.

T

-tbl

Syntax `-tbl`

Description Assigns a path to the tbl-files (filter).

Comment Is comparable to `-prn`

-tcinterval

Syntax `-tcinterval <days>`

Where *days* is the number of days between the reports.

Description Enables the use of the Transaction Counter in the StreamServer. The Transaction Counter has been replaced by the Communications Reporter in version 4.1.2 of the StreamServer.

Example `-tcinterval 2`

-td

Syntax `-td <path>\<directory>`

Description Specifies the path and name of an alternative temporary directory for a Project.

Comment The default temporary directory (`tmp`) resides in the working directory and can grow unrestrictedly. Use this argument to specify an alternative temporary directory outside the working directory.

Note: Specify a directory that the StreamServer can easily and quickly access.

Example `-td D:\StreamServe\TemporaryDirectory`

-timer

Syntax

`-timer (hex value)`

Where *(hex vale)* is the hex value for the timer you want to start. See the hex values table below for details of the timers and hex values.

Description

Times phases in the StreamServer.

Examples

`-timer 0x04` times how long the PreProc phase takes.

`-timer 0x14` times how long the PreProc and output connectors take.

`-timer 0xFFFF` starts all timers and writes the results to file.

Timers and hex values	
Input queue	0x01
Collect	0x02
PreProc	0x04
Process	0x08
Output Connector	0x10
Output Queue	0x20
Job	0x40
Write to file	0x80
Loadable external filers	0x100
All timers	OxFFFF
Script scopes – Measures time spent in script scopes, such as retrieved script, before Process script, etc.	Ox200
Script function CallProc – Measures time spent for each call to the script function CallProc.	Ox400
Script function Execute – Measures time spent for each call to the script function Execute.	Ox800

-tmpcompress

Syntax `-tmpcompress`

Description Compresses certain temporary files during the collect/preprocess/output sort phase.

Comment More CPU resources are used when using this argument, but less writing to disk is performed.

This means if you have really fast CPUs and slow disks, this argument can increase performance considerably. In the unlikely case that you have slow CPUs and fast disks, it could do nothing, or decrease overall performance slightly.

If a system's disk cache has a high hit rate, and does not fill up, this argument is usually unnecessary. This is often the case with realtime jobs.

If, during times of peak load, disk utilization is high but CPU utilization is low, then this argument will usually help. This is often the case with batch jobs, particularly large ones.

V

-v

Syntax `-v`

Description The StreamServer prints its version on standard error.

-var

Syntax `-var <variable_1>=<variable_2>`

Description An opportunity to create a variable when starting the server.

Comment Read-only. Cannot change later.

Example `-var dest=file.txt`

W

-wsin

Syntax `-wsin <file>`

Description Records a list of all incoming data that the StreamServer identifies. The list is recorded in the specified file.

Use the `-wsin` argument to produce a file containing a copy of the Message, showing the actual data in the Fields and Blocks that you have specified, and in the order in which StreamServe will process them.

Comment This argument is used during the testing phase since it is able to copy everything that will be processed in a file(s).

X

-xsdimport

Syntax `-xsdimport 0|1`

Description Used together with XMLIN Events and XML schemas. Specifies when to load the XML schemas defined in the namespace/schema location mapping table. Any settings in the mapping table will override `-xsdimport`.

0: load the schemas at StreamServer start-up.

1: load a schema when receiving input related to the schema. The schema must be read and parsed before validation.

Example `-xsdimport 1`