StreamServe™ Persuasion™

StreamServe Persuasion SP5 Communication Reporter

User Guide

Rev A

StreamServe Persuasion SP5 Communication Reporter User Guide
Rev A

StreamServe Web Site
http://www.streamserve.com

Contents

About Communication Reporter

Communication Reporter monitors how and which business documents are communicated within an organization and to its customers. With knowledge about what is communicated to customers, and how, companies can improve their interactions and bring them closer to their customers. The StreamServe Communication Reporter enables companies to set benchmarks to analyze trends in their communications, as well as compare customer interactions between different regions, countries, business units and departments.

Improved customer communication

Effective customer communication is based on knowledge. You need to know how and what you communicate to your customers in order to improve it. With Communication Reporter, you can monitor the types of business documents (e.g. invoices, special offerings, orders, etc.), formats (e.g. PDF, print), and channels (e.g. print, email, fax) used to communicate with your customers.

Cost savings/Efficiency improvements

The statistics produced by Communication Reporter can assist in analyzing communication channels. For example, what kind of savings could be expected if ten percent of traditional mail communications were electronic?

Departmental charging

Many companies run on tight budgets and promote consistent treatment of costs. Identifying costs is cumbersome. With Communication Reporter, you can monitor and identify the usage of resources (e.g. printers). Usage statistics are well-presented and automatically communicated to interested parties. Statistics can be used for inter-departmental billing or to improve operational efficiencies.

Per-transaction/Document licensing

Communication Reporter can be used to monitor and create comprehensive usage reports for usage-based pricing.

How it works

Communication Reporter offers the ability to count and monitor the usage of StreamServer and gathers data on key objects:

- Connectors, e.g. Email, Spool, FTP
- Processes, e.g PageOUT
- Drivers, e.g. PDF, PCL, AFP
- Total number of physical pages produced as output

You can define counters for specific objects other than the key objects. See *Defining counters for specific objects* on page 16.

Statistical reports can be created in XML and PDF format.

Required components

The following main components are involved when using Communication Reporter for creating usage statistics reports:

- The StreamServer on which usage will be monitored.
- The statistics repository in which usage data is stored.
- Reporter that retrieves the usage statistics and generates an XML file.
- A Project for producing and distributing usage reports.

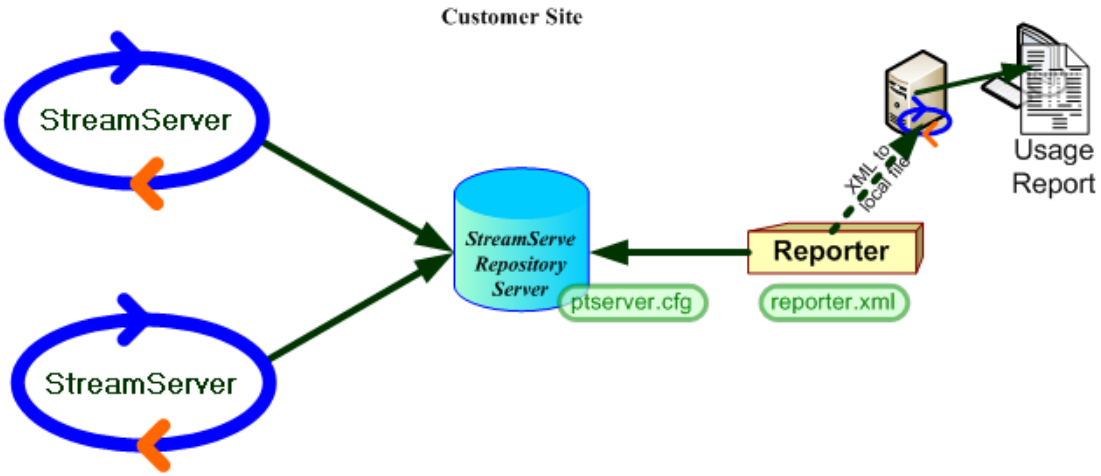


Figure 1 Communication Reporter

StreamServer, the usage statistics repository and Reporter can run on the same or on separate physical machines. They are all included in the standard StreamServe setup.

StreamServer

When a job is completed, StreamServer writes the usage statistics to the Repository. Multiple StreamServer instances can share the same Repository.

For information about what data StreamServer collects, see *How usage data is gathered in StreamServer* on page 9.

The usage statistics repository

StreamServer automatically creates a new database the first time the usage statistics repository is used. When you update a Project, StreamServer automatically adds new counters for components that do not already exist in the repository. Time stamps are added when new counters are created.

You only need one repository instance. StreamServers at remote sites, for example across a WAN, can access a central repository. Multiple repository instances are supported, but not required.

Reporter

Reporter queries the statistics repository daily, retrieves unreported usage data, and generates an XML file.

The XML file is stored in a directory from which another StreamServer instance can retrieve the file and generate a detailed usage report.

You cannot run multiple Reporter instances on the same machine. Reporter is not a critical component in the sense that if it goes down, StreamServer will continue to process jobs.

The template Project for producing and distributing usage reports

A template Project for producing PDF and XML formatted reports is provided with Communication Reporter setup. You can modify this Project according to your needs, for example if you want to distribute the reports via email.

See *Producing and distributing usage statistics reports* on page 20.

Security

The integrity of usage statistics is protected by checksum information. Reporter monitors these checksums for tampering or corruption. If bad checksum information is detected, Reporter sets a data corruption warning flag.

What data is stored in the statistics repository

There are two main categories of information that will be stored in the statistics repository:

- System information
- Usage information

System information

The following information is written or updated in the repository when StreamServer starts:

- The customer number retrieved from the license file.
- A unique ID for the StreamServer instance, which you define in the Platform configuration.
- A unique report ID, composed of the StreamServer instance ID and a generated serial number.
- The StreamServer version.
- Date and time of the first created counter, the last updated counter and when the report was created.

Usage information

Usage information is written upon job completion and identifies usage of input and output components.

For input data, the following information is written to the repository:

- Input connector name and type.
- Event name and type.
- Message name.
- The value of the counter.

For output data, the following information is written to the repository:

- Process name and type.
- Driver type.
- Output connector name and type.
- Total number of produced physical pages.
- The value of the counter.

Note: Statistics about Collector usage can not be collected.

How usage data is gathered in StreamServer

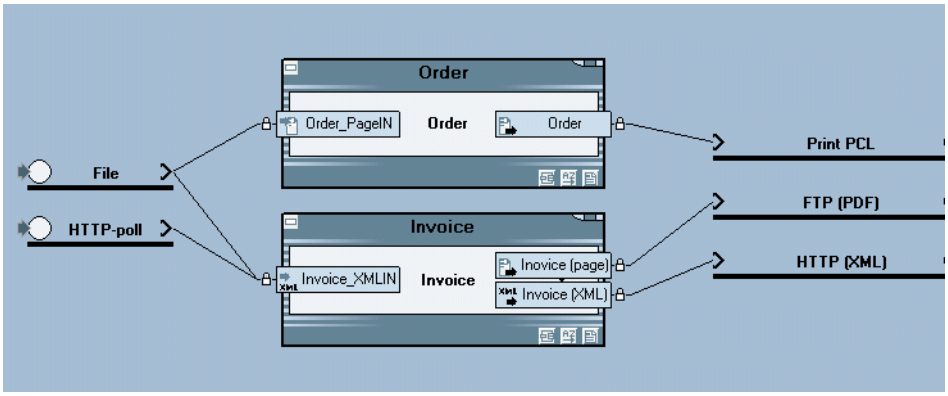
StreamServer updates input and output counters for each unique chain of components used when processing jobs:

- Input is received by an input connector, parsed by an Agent, and the result is written to a Message.
- A Process uses the content of the Message to create the output, which is then passed through a driver that creates the output format. The output is delivered by an output connector.

When a job is processed by StreamServer, the number of unique chains is counted. For each chain, the name, type and product ID of each component used in the chain are stored. Additionally, the number of times each chain was used is also stored.

For example, assume that you have a Project with the following components:

- Input connectors: File and HTTP Poll
- Events: PageIN and XMLIN
- Messages: Order and Invoice
- Processes: Order, Invoice (page) and Invoice (XML)
- Drivers: PCL and PDF (no driver is used for XMLOUT)
- Output connectors: Print PCL, FTP (PDF) and HTTP (XML)



An input sequence consists of one input connector, one Event and one Message. In this example, data is gathered based on the following input sequences:

Input connector	Event type	Message
File	PageIN	Order
File	XMLIN	Invoice
HTTP-poll	XMLIN	Invoice

An output sequence consists of one Process, one driver and one output connector. In this example, data is gathered based on the following output sequences:

Process	Driver	Output connector
Order	PCL	Print PCL
Invoice (page)	PDF	FTP (PDF)
Invoice (XML)		HTTP

Performance

In most cases, Communication Reporter will have no significant effect on overall performance.

If you have high-volume real-time processing requirements with thousands of transactions per minute and per CPU, you should evaluate performance when running Reporter as part of the overall performance evaluation and tuning, before taking your Projects into production mode.

Repository server load, performance, and disk storage are highly dependent on the rate of job completion and the number of StreamServers reporting statistical information. A Repository Server that needs to log tens of thousands of jobs per minute should run a dedicated statistics repository on a separate physical disk. Depending on performance requirements, the statistics repository may have to reside on a separate physical machine under extremely high loads.

Configuring the Communication Reporter

The same system and user requirements apply as for StreamServer, see the *StreamServe Installation Guide*.

The StreamServer, the repository server and Reporter application can be installed on the same physical machine, or on separate machines. See also *Performance* on page 11.

Requirements

To enable Communication Reporter to gather usage data, you must have a valid license.

To use Communication Reporter

To use Communication Reporter you must do the following:

- Configure the StreamServer to be monitored, see *Configuring the StreamServer to be monitored* on page 15.
- Configure Reporter, see *Configuring Reporter* on page 17.
- Configure the Project for producing a statistics report. Use the template Project provided with Communication Reporter installation, see *Producing and distributing usage statistics reports* on page 20.

Installing Reporter

Reporter is included in the standard StreamServe installation.

Windows installation

After a successful installation on Windows, the Windows service `StreamServer Reporter` is added. You start/stop this service the same way as you start/stop other Windows services.

UNIX installation

After a successful installation on UNIX, all Reporter components are added to

`applications/usr`

To start Reporter, you must run the following command:

```
$ ./streamserve usr
```

Configuring the StreamServer to be monitored

To enable StreamServer to gather usage data and store it in the statistics repository you must do the following:

- Specify the repository in which StreamServer will store usage statistics, see *Specifying the repository for storing usage data* on page 15.
- Specify whether StreamServer should reconnect to the statistics repository in case of a network failure. See *Enabling repository server reconnection* on page 15.

You can define counters for specific objects, other than the objects counted automatically. See *Defining counters for specific objects* on page 16.

Specifying the repository for storing usage data

You must specify the location of the repository in which StreamServer will store the collected usage data by adding the `ProcessingStatistics` keyword to the Platform. You can also specify a unique StreamServer ID to identify usage data from different StreamServer instances. The ID will be shown in the report.

Syntax

```
ProcessingStatistics "<ServerInstanceID>" "<Hostname>" end;
```

where:

- *<ServerInstanceID>* is an optional unique name for the StreamServer. The ID will be shown in the report.
- *<Hostname>* is the name of the machine on which the repository server runs. If StreamServer and the repository server run on the same machine, enter localhost.

  `LOCAL` is not allowed since Reporter also accesses the repository.

Example: `ProcessingStatistics "user01" "localhost" end;`

To specify the repository location

See *Using custom commands and keywords* in the *Design Center* documentation for information on how to add custom commands and keywords to the Platform.

Enabling repository server reconnection

Specify whether StreamServer should reconnect to the repository server in case of a network failure by adding the `DefaultRepositoryFailover` keyword to the Platform.

Syntax

```
DefaultRepositoryFailover <Retries> <Sleep> <Closedown>;
```

where:

- `<Retries>` is the number of reconnection attempts. When set to `-1`, StreamServer will try to reconnect until a connection is successfully established.
- `<Sleep>` is the time (in seconds) between reconnection attempts.
- `<Closedown>` defines whether StreamServer will continue to run if a reconnection attempt fails:
  - If set to `0`, StreamServer will continue to run.
  - If set to a value other than `0`, StreamServer shuts down if the last reconnection attempt fails.

Example: `DefaultRepositoryFailover 1 15 1;`

To enable server reconnection

See *Using custom commands and keywords* in the *Design Center* documentation for information on how to add custom commands and keywords to the Platform.

Defining counters for specific objects

You can define a counter for a specific object in the Project. When the job is complete, the counter is saved in the repository thus enabling it to be reported by Communication Reporter.

Defining counters for specific objects can be useful if you want to count, for example:

- specific XML elements
- specific StreamOUT fields
- information retrieval from databases.

To define counters for specific objects, and to increase the value of the counters, you use the `IncProcStatCounter` script function. See the *StreamServe Scripting Reference*.

Configuring Reporter

You must edit the Reporter configuration file and specify the following:

- The repository from which to retrieve usage data.
- When to retrieve data from the repository.
- Where to store the generated XML file.

Running Reporter and the repository on different machines

If you have installed Reporter on a machine other than the repository server, you must make sure Reporter uses a domain account with access to the repository server. To enable Reporter to retrieve data from the repository, you must specify the repository server name, user name and password in the configuration file, see *Reporter configuration attributes* on page 18.

The Reporter configuration file

The Reporter configuration consists of an XML file located in the installation directory:

<reporter_installation_directory>\bin\usr.xml

Use a text editor to edit the file. You must restart Reporter for changes to take effect.

Example 1 *Example of a Reporter configuration file*

```
<?xml version="1.0"?>
<configuration xmlns="http://www.streamserve.com/strs/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <reporter>
    <repository>
      <server name="localhost" />
      <user name="" password="" />
      <reconnect attempts="3" delay="5" />
    </repository>
    <security configfile="./security.ssc" />
    <scheduler start="1" interval="3" forcesend="false"/>
  </reporter>
  <receiver>
    <proxy name="" port="" />
    <destination url="file://C:\dailyXML\report.xml"
                    clientsecurity="SSL_CLIENT" />
  </receiver>
</configuration>
```

Reporter configuration attributes

Repository connection	
`server`	`name` – The name or IP address of the machine on which the Repository Server runs. Default value is `localhost`.
`user`	`name` – The user name for accessing the repository server. `password` – The password for accessing the repository server.
`reconnect`	`attempts` – The number of times Reporter tries to reconnect to the repository in case of a failure. The default value is `3`. `delay` – The number of seconds Reporter waits until the next reconnection attempt. The default value is `5`.

Data retrieval and report destination	
`scheduler`	Reporter retrieves and sends the report at a random time within a specified time window. You specify when this time window begins (`start`) and its duration (`interval`). For example, if the report should be sent between 23:00 and 02:00, you specify `start="23"` and `interval="3"`. `start` – The time at which Reporter will start to retrieve data from the repository and send the report. Valid range is between `0` and `23`. `interval` – The duration (in hours) of the time window within which Reporter must retrieve and send the report. Valid values are `3` or higher. Default value is `3`. `forcesend` – When set to `true`, Reporter is forced to retrieve and send the report immediately at startup. Default value is `false`.

Data retrieval and report destination	
`destination`	`url` - The destination path to which to send the statistics report. You can specify one or more paths separated by a semicolon (`;`). If you are using the template Project included in Communication Reporter installation to consolidate and distribute reports, the destination path must match the input directory scanned by the `DailyXMLReportDirectory` connector, see *Producing and distributing usage statistics reports* on page 20.

Producing and distributing usage statistics reports

Communication Reporter installation includes a template Project to help you get started with creating usage statistics reports. The Project is located in the installation directory:

`<reporter_installation_directory>\bin\Communication Reporter Project.dcpackage`

The Communication Reporter Project scans the directory in which daily XML reports are stored and consolidates the daily reports into a single report. The consolidated report can then be formatted and distributed, for example a PDF file can be created and distributed via email.

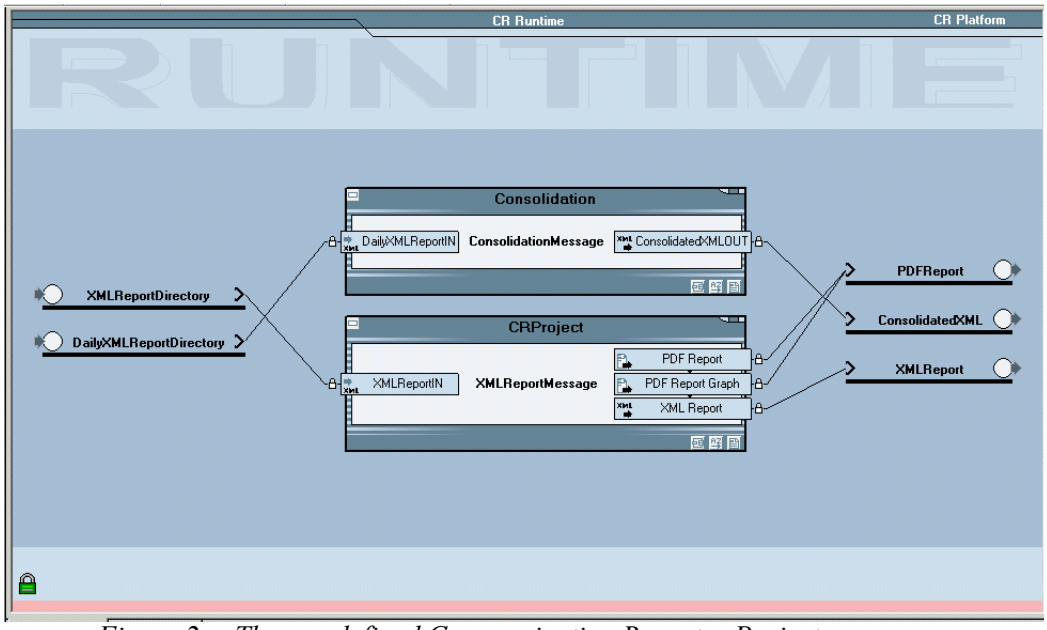


Figure 2 The pre-defined Communication Reporter Project.

What you need to configure

To use the template Project, you need to do the following:

- Configure the `DailyXMLReportDirectory` input connector to scan the directory in which the daily XML files are stored, i.e. the directory specified by the `destination` parameter in the Reporter configuration file, see *Reporter configuration attributes* on page 18.
- Schedule when to produce consolidated reports by defining a polling interval for the `DailyXMLReportDirectory` input connector.
- Specify the directory in which the `Consolidated XML` output connector will store the consolidated report, and configure the `ConsolidatedXMLReportDirectory` to retrieve the report from this directory for further processing.

- Configure the `Create Presentation Report` Message as required for formatting and distributing the statistics reports, for example as a PDF file via email, or for storing the reports in a database.

The mapping file

The comprehensive amount of usage data collected by StreamServer can be combined in many different ways. For example, if you have one PCL and one Postscript connector, and you want to see the number of Print category items that were produced, you must consolidate these two into a single Print category. This is done using a mapping file.

The pre-defined Project has a mapping file called `processTypeColumn.tbl`, which is stored in the resource set connected to the Platform. The `processTypeColumn.tbl` file contains the most commonly used combinations of print, email and fax output.

You can change this file, for example the category names (print, email, etc.) to better suit your naming conventions, or if you add new components to the Project you are monitoring.

Example 2 Mapping file example

```
//!CodePage UTF8!
10,1,PCL chdef_print
54,1,PDF chdef_archive
50,1,PDF chdef_email
```

Log messages

Communication Reporter writes to the `log.txt` file, located in the `Reporter\bin` directory.

Communication Reporter log messages	Cause
`Invalid application configuration: <variable>.`	Incorrect settings in the configuration file. Check the `report.xml` file according to the node specified by *<variable>*.

Usage statistics log messages generated by StreamServer are written in the StreamServer log file.

StreamServer log messages	Cause
`4857 Warning - Failed to add statistics for process (<errorCode>, <databaseSpecificError>, <connector>, <connectorType>, <processName>, <processType>, <execTimes>, <pageCount>).`	The statistics repository can not be accessed, due to, for example, stopped repository server or network failure.
`4858 Warning - Failed to add statistics for message (<errorCode>, <databaseSpecificError>, <connector>, <connectorType>, <messageName>, <eventType>, <count>).`	The statistics repository can not be accessed, due to, for example, stopped repository server or network failure.
`4859 Error - Processing statistics database must not use LOCAL mode.`	The repository server is set to run in LOCAL mode. Change the settings for the `ProcessingStatistics` keyword, see *Specifying the repository for storing usage data* on page 15.
`4897 Error - Failed to update processing statistics system information. Error %d1 (%d2).`	The statistics repository can not be accessed, due to, for example, stopped repository server or network failure.
`4903 Error - The server instance id must not be empty.`	The *<ServerInstanceID>* for the `ProcessingStatistics` keyword is missing or incorrect, see *Specifying the repository for storing usage data* on page 15.
`4917 Error - Failed to start up statistics collecting. Make sure that the reporter application has been started.`	Reporter is not running. Start the service in Windows, or process in UNIX.

24 Log messages