SoftPI – Software Projects & Innovations

SoftPI Flow Collector

User guide

Version: 2.00

Date: May 2012

Introduction 3
1. Features and benefits.....3
2. Software components.....4
3. Hardware and software requirements.....5
4. Installation.....5
5. Uninstallation.....6
6. Configuration.....7
6.1. Settings of data collection parameters.....7
6.2. Settings of data storage.....9
6.3. Settings of aggregation parameters.....12
6.4. Installation and start of the SoftPI Flow Collector service.....13
7. Service monitoring.....13
8. Reports.....15
Attachment. List of fields.....18
Technical Support.....25

Introduction

SoftPI Flow Collector system (previously named SoftPI NetFlow Collector) provides to collect information about network flows using NetFlow versions 5 or 9 (Cisco Systems), RFlow and IPFIX (RFC 5101, 5102), as well as flexible aggregation of collected data with storing theirs in a storage of one of types:

- Microsoft SQL database (2000, 2005 or 2008 editions),
- MySQL database,
- text file.

System works under Windows XP/2003/Vista/2008/7 (Microsoft).

The sources of network flows in NetFlow, IPFIX, and RFlow can be: routers, wireless access points, switches and other network devices, as well as computers running any operating system with software network sensors.

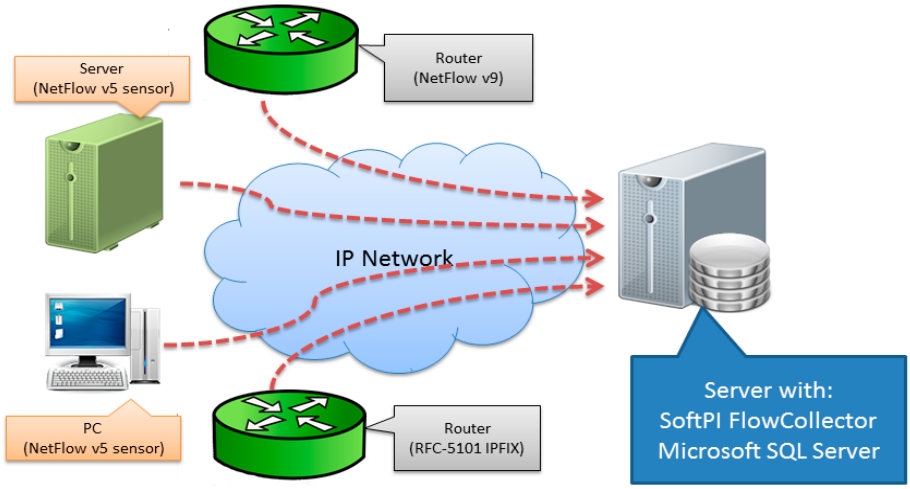

Figure 1

As shown in Figure 1, SoftPI Flow Collector is installed on the server. Microsoft SQL Server can be used as data storage. As a storage can be used also MySQL or text file. Flow information come from routers with NetFlow version 9 and IPFIX, and computers where the network sensor with NetFlow version 5 is installed.

Figure 1 shows only one example of using SoftPI Flow Collector. Other configurations are possible.

1. Features and benefits

Supported formats and protocols

SoftPI NetFlow Collector supports collection and parsing of the following network protocols:

- NetFlow version 5;
- NetFlow version 9;
- RFlow;
- IPFIX.

All fields of these protocols are supported. User can select only the required fields to preserve in the database.

Flexible aggregation and storage

SoftPI Flow Collector provides configuration of:

- Custom list of fields which are stored.
- Custom list of fields for aggregation.
- Custom list of IP address of devices which are sources of network traffic data.
- Separate log for each devices that work as sources of NetFlow, IPFIX, or RFlow.
- The possibility to collect data over multiple IP ports.

Supported types of storages

The system provides the following types of storages:

- text file,
- Microsoft SQL Server 2000/2005/2008,
- MySQL server.

GUI to make configuration and build reports

Configuration of SoftPI Flow Collector is performed in easy-to-use graphic interface.

No restrictions on the number of network devices

SoftPI Flow Collector has no restrictions on the number of network devices with which it can collect information. Restrictions may arise only due to insufficient computer resources.

Processing of information about network traffic

To process information from the storage, the user can use any available software tools supporting the relevant storage.

Possibility of forming reports on network activity is built in SoftPI Flow Collector. This possibility may be available if the user uses Microsoft SQL Server 2008 R2 and above as a storage and uses set of fields which is a similar to set of fields of NetFlow version 5.

User can edit available reports using Microsoft SQL Server Reporting Services Report Builder 3.0, or create your own. Reports, that are part of the system, or by yourself can be accessed through the custom Web-site using SQL Server Reporting Services (SSRS).

2. Software components

The system contains the following components:

- **SoftPI Flow Collector** - a flow collector service;

- **Flow collector administration** program. The program provides a graphic interface to configure Flow Collector parameters, as well as allows to run and stop the Flow Collector service.

Built-in reporting system with 9 predefined reports. Reporting system is based on Microsoft Reporting technology.

3. Hardware and software requirements

Minimum hardware requirements for a computer, where SoftPI Flow Collector will be installed:

- Processor: 1 GHz or more.
- RAM: 512 MB or more.
- Hard disk: 40 GB.
- LAN port: 100 or 1000 Mbit/sec.

Software requirements for a computer:

- Windows XP/Vista/7 or Windows Server 2003/2008;
- .Net Framework 3.5. If computer where you install SoftPI Flow Collector has an Internet connection and does not contain .Net Framework 3.5, then .Net Framework 3.5 is downloaded and installed automatically. If the computer at the time of installation of the system will not have a connection to the Internet, then to work the system, you must download and install .Net Framework 3.5.
- Depending on the intended type of storage may need to install: MySQL 5.0 or Microsoft SQL Server 2000/2005/2008 or 2000MSDE/2005 Express/2008 Express.

4. Installation

Run the install file. The window of installation master appears (Figure 2).

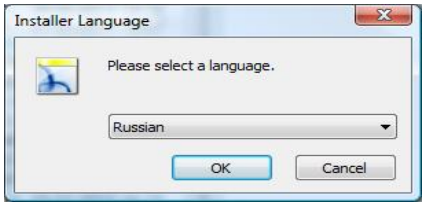

Figure 2

Select the required language. There are options:

- Russian
- English.

Click **OK**.

In the appeared window, click **Next** to continue installation.

In the next window (Figure 3) you can change the folder where SoftPI Flow Collector will be installed.

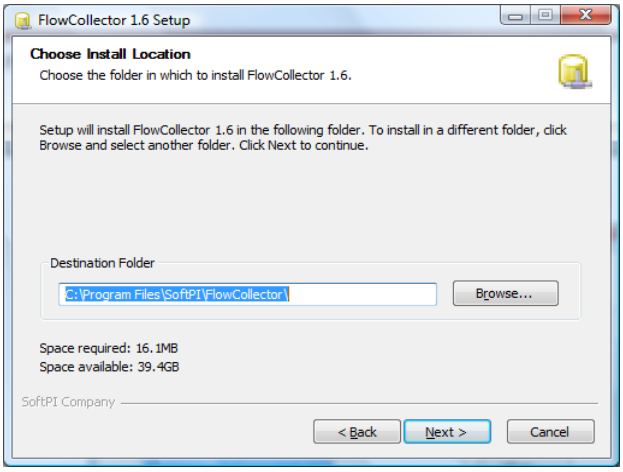

Figure 3

By default, the folder is used: ...**\Program Files\SoftPI\FlowCollector**
Click **Next**, the window appears as shown in Figure 4. You can change the program name that will be used in the **Start Menu**. By default, the **SoftPI FlowCollector** name is used.
Click **Install** to install SoftPI Flow Collector.

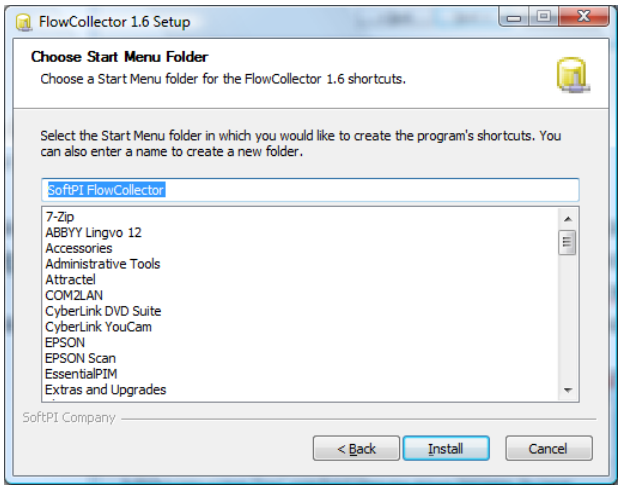

Figure 4

5. Uninstallation

To uninstall the SoftPI Flow Collector:

- Open **Control Panel**.
- Select "**Programs and Components**" ("Add/Remove Programs").
- Select the **Flow Collector** item and click the "Change or Remove Programs" button.

Note

When you uninstall the program, the data storage is not deleted. If you use Microsoft SQL Server or MySQL as a data storage, apply the appropriate tools to delete the storage.

6. Configuration

Configuration and run of the **SoftPI Flow Collector** service are performed in the **Flow collector administration** program and this process consists from the following steps:

- setting the parameters of data collection (the **Traffic collector** configuration page);
- you must decide what type of storage will be used. If Microsoft SQL Server or MySQL will be used as storage, you must download the appropriate installation of SQL server and install it on your computer*;
- setting the parameters of data storage (the **Storage** configuration page);
- setting the parameters of aggregation if it is necessary (the **Aggregation** configuration page);
- start of the service (the **Service** configuration page).

* In case you choose to use Microsoft SQL Server 2008 R2 as storage, you can use advices of Microsoft for installing software or our advices from the article: http://www.tariscope.com/en/support/knowledge-base/9-tariscope-3x/66-sql-server-2005-2008-install.html

6.1. Settings of data collection parameters

Start the **Flow collector administration** program and click **Traffic collector**. The program takes the form as shown in Figure 6.1.

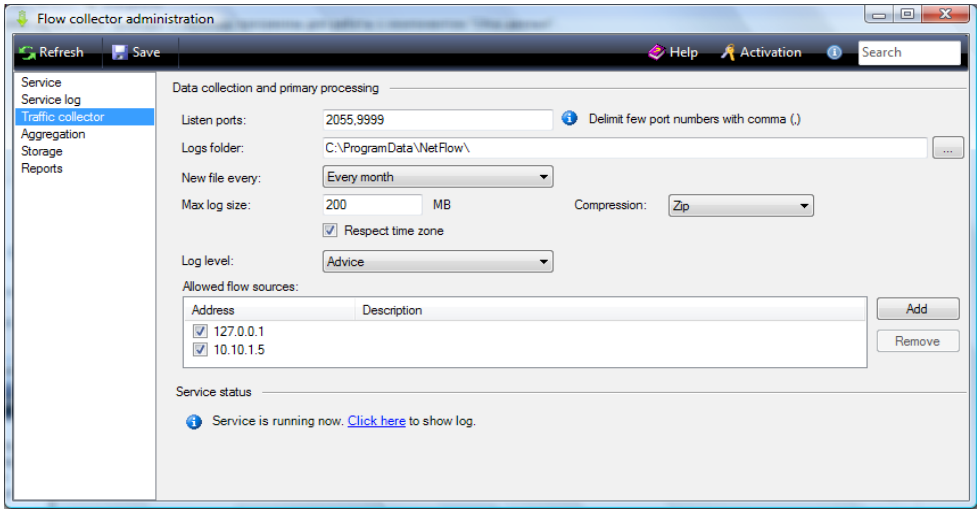

Figure 6.1

In the **Listen ports** box, enter a list of IP ports, from which the Flow Collector service will collect data. It should be borne in mind that the service works using only UDP. In case using more than one IP port you must enter their numbers using comma. By default is set to 2055. This port is usually used for NetFlow. For IPFIX protocol according to RFC 5101, port 4739 should be used to listen for an unsecure connection. However, not all manufacturers adhere to this requirement. So you should accurately determine the number of IP port used for data transmission by NetFlow, IPFIX or RFlow protocols in the documentation for telecommunications equipment.

Flow collector except processing the incoming data flow and writing it in storage executes the backup of the data flow into a log file in the form in which these data came from the network interface.

In the **Logs folder** box you can change the path to the folder where the files will be stored. The default is the folder: **\ProgramData\NetFlow**

In the **New file every** list, select a required period to create a new log file. Possible options are:

- **Don't rotate**;
- **Every hour**;
- **Every day**;
- **Every month**.

Regardless of the selected period in this list you can determine the size of the log file in the **Max log size** box. After reaching the set value a new log file will be created. The default size is 200 MB.

To reduce the volume occupied by the log file, you can perform a compression. Compression is automatically performed immediately when data is written to the file. You can use different data compression algorithms that determined in the **Compression** list. The **Compression** list contains the following options:

- **No compression**;
- **Zip**;
- **Bzip**;
- **Zlib**.

The SoftPI Flow Collector service can log the results of their work with varying degrees of detail. The level of detail is determined by the parameters specified in the **Log level** list. There are options:

- **Status**,
- **Critical error**,
- **Error**,
- **Warning**,
- **Information**,
- **Advice**,
- **Debug**.

The **Status** level is the least detailed level. The **Debug** level is the most detailed level, for example, the IP addresses of devices from which data is received are displayed. The default is the **Information** level.

Specify IP addresses of telecommunications devices, from which the Flow Collector service must collect data, in the **Allowed flow source** table. To add a new IP address, click on the **Add** button. The **Flow source** window appears as shown in Figure 6.2.

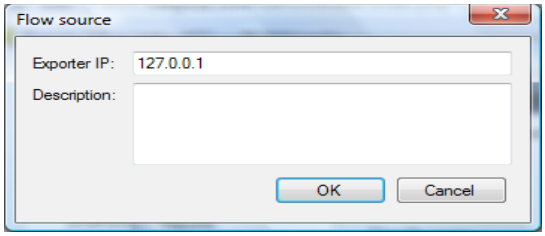

Figure 6.2

Enter IP address of the required device in the **Exporter IP** box.

The **Description** box is for informational purposes and is not required for entry.

***Note**. The SoftPI Flow Collector service can itself add IP addresses of telecommunications devices, from which data is received, to the **Allowed flow source** table. But if a specific device is not selected in the table, the data from that device will not be processed. Accordingly, for data processing with the required data sources, select the required IP address.*

After entering all the above parameters, click on the **Save** button located on the toolbar.

6.2. Settings of data storage

As mentioned above, before settings of storage parameters the user must decide what type of storage will be used. If a storage is supposed to use Microsoft SQL Server or MySQL, you should download the appropriate installation and install on your computer.

We recommend to use Microsoft SQL Server 2008 R2 as a storage. By using this server as a storage and choice of fields to store information about network flows similar to the fields of NetFlow version 5, the user will be available the **Reports** page from the **Flow collector administration** program. Otherwise, the user must himself find an application that will provide the required reports from the storage.

Click **Storage**. The program will take a form similar to that shown in the Figure 6.3.

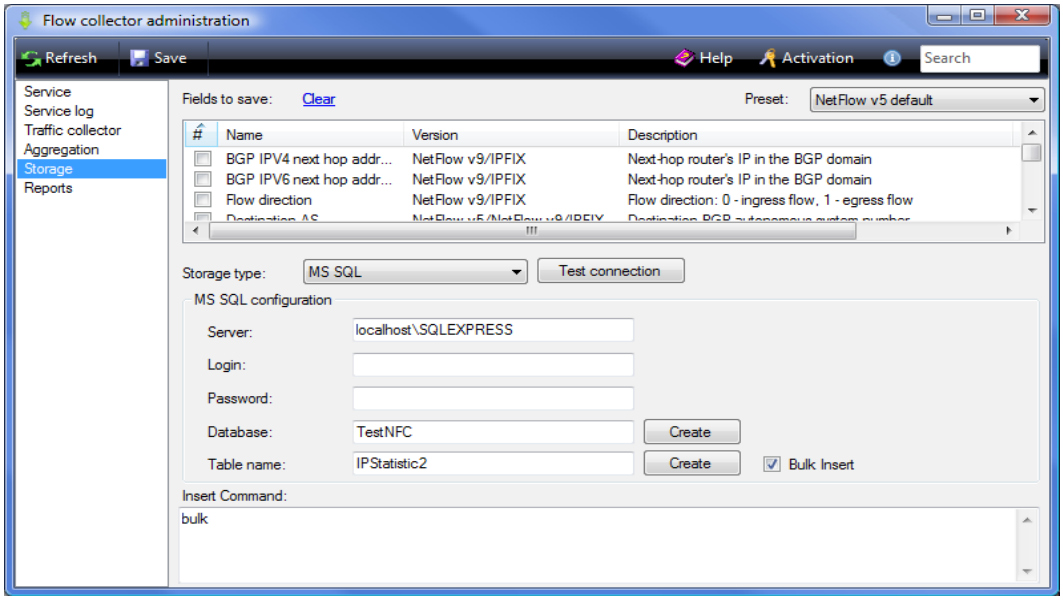

Figure 6.3

Define a list of fields that will be processed and stored in the data storage. You can use a predefined set of fields. Choice of predefined set of fields. is performed from the **Preset** list.

You can select the following options:

- **NetFlow v5 default** – provides choice of the most commonly used fields of NetFlow version 5;
- **NetFlow v9 default** - provides choice of the most commonly used fields of NetFlow version 9;
- **IPFIX default** - provides choice of the most commonly used fields of IPFIX;
- **All NetFlow v5 fields** - provides choice of all fields of NetFlow version 5;
- **All NetFlow v9 fields** - provides choice of all fields of NetFlow version 9;
- **All IPFIX** - provides choice of all fields of IPFIX.

If the user is interested in any particular set, which does not correspond to any of the above sets, he can select the desired fields by checking the required rows of table in the "#" column.

If you plan to receive data from multiple network devices that use different protocols, respectively, select the fields that are supported in all types of required protocols.

The table of fields contains the following columns:

- shows selected or not, this field;

Name — displays the name of the field;

Version — displays type and version of protocol;

Description — a brief description of the field;

Field code — the field name that is used in the database. The name is also used in the **Insert command** box for the automatic creation of a database query to store data.

The table supports sorting information. To do this, click on the name of the column of interest.
Right-click on the bar table with the names of the columns leads to a menu, similar to that shown in Figure 6.4.

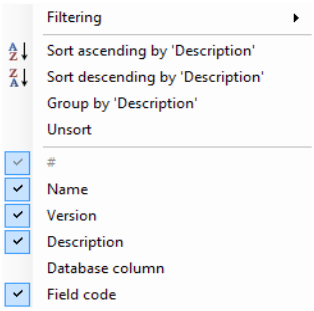

Figure 6.4

"**Filtering**" - Selecting this menu item leads to an additional menu, where the user must check the letters, which are the first in names of any of the fields. This leads to a corresponding filtering information.

"**Sort ascending by 'XXXXX'**". It allows to sort ascending by 'XXXX' field. The selected 'XXXX' field is determined by the name on any of the columns in the cursor was at the time to click for the menu.

"**Sort descending by 'XXXXX'**". It allows to sort descending by 'XXXX' field. The selected 'XXXX' field is determined by the name on any of the columns in the cursor was at the time to click for the menu.

"**Group by 'XXXXX'**". Allows to group data in the table on the first letters of words in the selected column.

"**Unsort**". This item disable previously the set grouping mode in the table.

To quickly find the desired parameter in the table we recommend to use the search mode. It is done by entering the required information in the position with the "**Search**" word on the right side toolbar.

In that case when you desire to use the **Report** page of the program you must:

- in the **Storage** type list, select the **MS SQL** option;
- select the fields for NetFlow version 5.

There are the following options in the **Storage type** list:

- **MS SQL** – Microsoft SQL Server 2000/2005/2008 is used as a storage. If you have not purchased Microsoft SQL Server and is not going to purchase, we recommend you use the free edition Microsoft SQL Server 2008 R2 Express.
- **MySQL** – MySQL Server is used as a storage.
- **CSV file** – A text file is used as a storage. The file must have the CSV format.

If you selected the **MS SQL** option, type a server name or its IP address (if it is necessary, enter IP address with a server name) in the **Server** box.

In the **Login** and **Password** boxes, type the user name and password, with which SoftPI Flow Collector will connect to SQL server. If you use Microsoft SQL Server on the same computer, where SoftPI Flow Collector will work, by default, Windows authentication will be used. In this case, a user name and password are not needed.

In the **Database** box, type a database name, in which data will be stored. When initially setting up the Flow Collector, a database should be created. To do this, click on the **Create** button, which is in the same line as the **Database** box. Upon successful completion of the operation to create a database, a message appears: "**Database 'xxxxxxxx' was created successfully**", where xxxxxxxx - a database name.

In the **Table** box, type the table name that will be created in the database. In this table the data will be stored. When initially setting up the Flow Collector, a table should be created. To do this, click on the **Create** button, which is in the same line as the **Table** box. A window appears as shown in Figure 6.5.

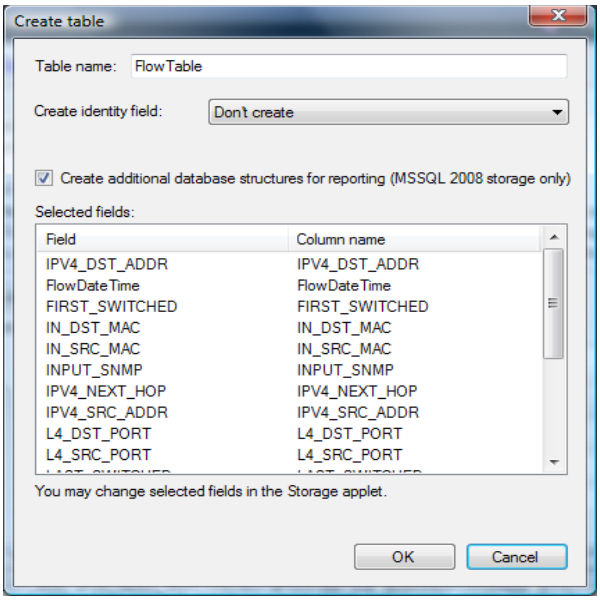

Figure 6.5

The **Table name** box displays the name of the table to be created.
The **Selected fields** table contains fields that user selected in the **Storage** configuration page.
The **Create identity filed** check box is used to set up a key field in the table. This key field can be used to select data on this field. There are the following options:

- **Don't create**. This option is used by default when you do not want to perform the data selection using a key field. The absence of a key field allows you to several reduce the size of databases.
- **binint**. The bigint format is recommended to specify if necessary in a key field, a relatively low traffic and the appropriate amount of information coming into the database.
- **GUID**. The globally unique 128-bit identifier. It is recommended to use it if necessary in a key field, high-traffic, which corresponds to the large volume of information coming into the database.

The **Create additional database structure for reporting** (MSSQL 2008 storage only) check box is used when user uses Microsoft SQL Server 2008 R2 and desire to have information about type of IP protocols and IP ports in the reports.

After choice of required parameters, click on the **OK** button. Upon successful completion of the operation of table creation, a message appears: "**Table 'xxxxxxx' was successfully created**", where xxxxxxx - the name of the table.

To verify that the specified parameters Microsoft SQL Server, click on the **Test Connection** button. In the case of a successful connection with SQL server message appears: "**Connection tested**". In case of failure to connect you should validate the entered parameters.

If you do not need any special handling of fields written to the database, and there is a large flow of data from NetFlow devices, check **Bulk Insert** box. In the **Insert command** box the "**bulk**" text appears. Using the the **Bulk Insert** command provides much more rapid data input into the database.

In case you chosen the **MySQL** option in the **Storage type** list, the **Port** list appears near the **Server** box. Select the required value of IP port from this list. By default is used: 3306.

In case you selected the **CSV file** option in the **Storage type** list, the program window partially changes form, as shown in Figure 6.6.

The **Filemask** box is used to configure a path to the file and a file name mask. To set a mask the following options are used:
%y% - year;
%m% - month;

%d% - day;
%dev% - IP address of device from which data are collected.

By default this file has the extension: log.

User can specify a delimiter in the **Column delimiter** box.

In the **Max size** box, type or select the required file size. Upon reaching the typed value of size, a new file is automatically created.

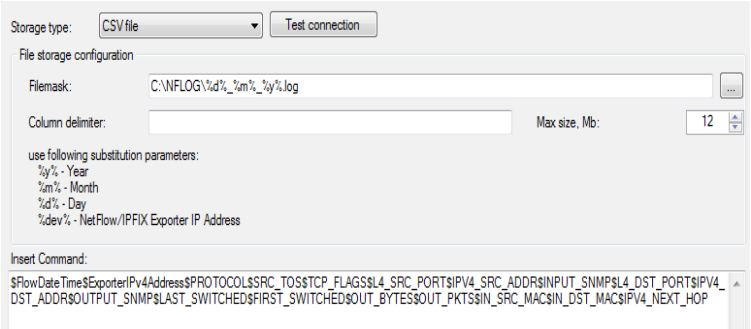

Figure 6.6

After set up all required parameters in the **Storage** configuration page, click **Save** on the toolbar.

6.3. Settings of aggregation parameters

If you plan to aggregate data on some of the field, you must configure the aggregation parameters. Using aggregation allows to significantly reduce the database size, but at the expense of losing the full details. To set up the aggregation parameters, select the **Aggregation** configuration page. The program window changes as shown in Figure 6.7.

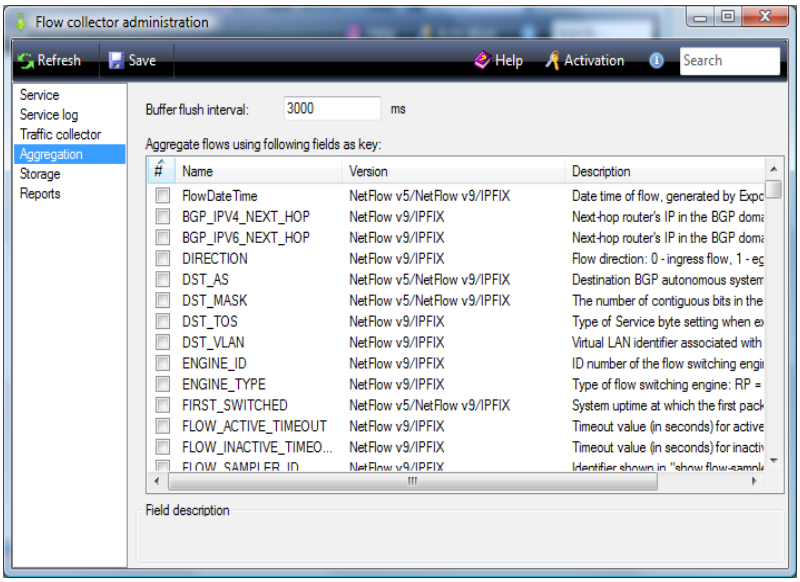

Figure 6.7

The user can set the aggregation of any of the fields specified in the table. The table contains the columns:

- #. This column is used to choose the required field.
- **Name**. Displays a field name.
- **Version**. Displays a protocol type and version.
- **Description**. Displays a brief description of field.

- **Data type**. Displays a data type of field.

By default, aggregation was set on multiple fields. This means that values of all fields for unique value of these fields will be aggregated during the period specified by the **Buffer flush interval** box. By default, the value of this parameter is 3000 ms. User can change this value. It should be borne in mind that reduction of the value leads to growth of the load on the CPU. Increasing the interval reduces the load on the processor, but the user during this interval may not have access to the data obtained during this period and, therefore, to analyze them.

If you specify multiple fields for aggregation, the aggregation will be performed for a set of specified fields.

To work with data tables (sorting, grouping, searching) apply the same methods described for the table of the **Storage** configuration page.

6.4. Installation and start of the SoftPI Flow Collector service

To install and start the service, select the **Service** configuration page. The program window for this page is shown in Figure 6.8.

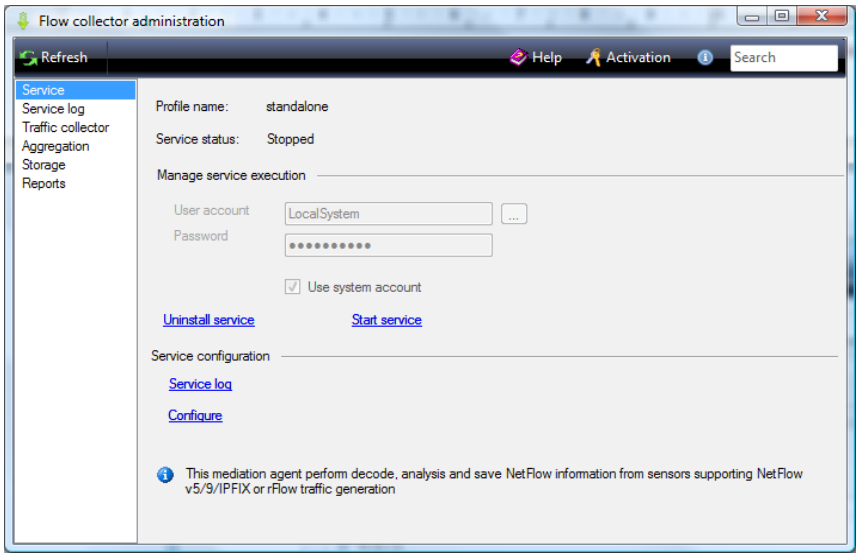

Figure 6.8

The **Service status** box displays the current service status.

Click on the **Start service** link to start the service. If start is successful the link will be replaced with **Stop Service** and the **Uninstall service** link will become inactive.

Click on the **Stop service** link to stop the service. After stopping the service the link changes its value at **Start Service** and the **Uninstall service** link will become active.

To keep track of the details of the SoftPI Flow Collector, click on the **Service log** link or select this feature in the list of program configuration pages.

Click on the **Configure** link, to move on the **Traffic collector** configuration page.

7. Service monitoring

To monitor the Flow Collector work, use the **Service log** page. If you select this page, program will look as shown in Figure 7.1.

Detailing of log entries are determined by value of the **Log level** list on the **Traffic Collector** configuration page.

For any detailing level the start time, main parameters, with which the service was started, error messages are displayed in the log.

If you select the **Debug** level in the **Log level** list, information about data sources are displayed in the log as shown in Figure 7.2.

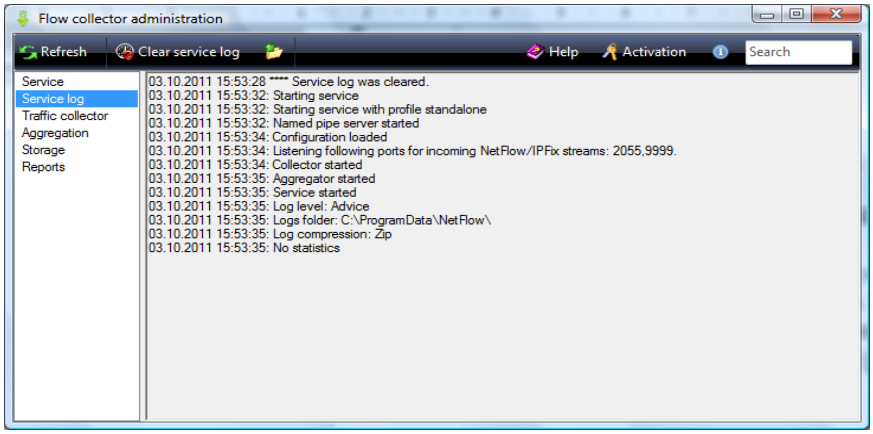

Figure 7.1

Therefore this level we recommend to apply at the stage of setting up system to make sure that the data from the required source is arrived in the collector.

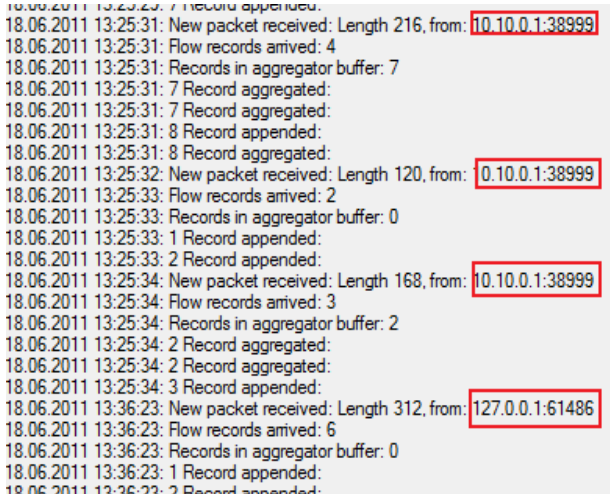

Figure 7.2

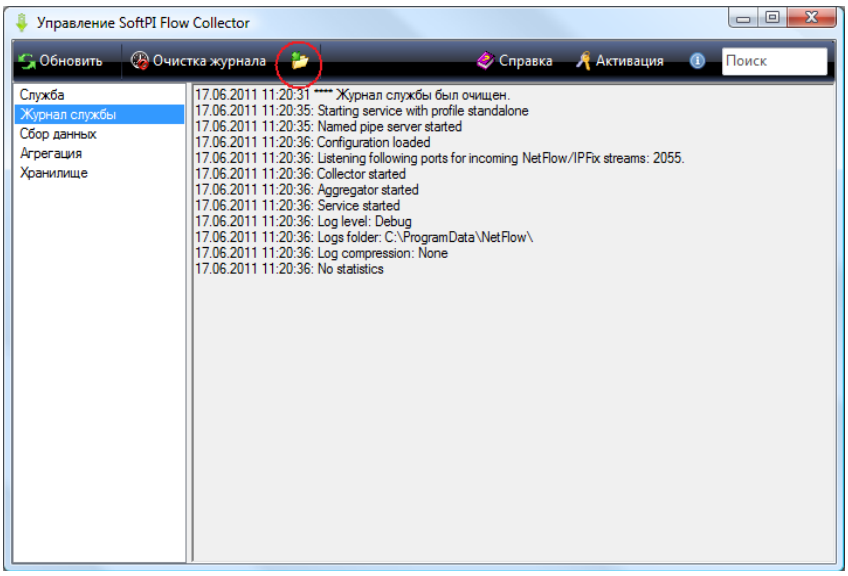

Figure 7.3

After setting up system, recommended to specify the **Information** log level. This logging level is sufficient for normal operation, but requires significantly less space than the **Debug** log level.

If you need to work with the log file, click on the icon, which encircled with a red circle in Figure 7.3. The folder with **standalone.log** file is opened.

To refresh information in the log, click **Refresh** on the toolbar.

To clean the log, click **Clear service log**.

8. Reports

The **Reports** page of the Flow collector administration program you can use only in case when:

- Microsoft SQL Server 2008 R2 is used as a storage,
- database fields correspond to fields of at least NetFlow version 5.

User can edit any report included in the system or create custom reports. Use the SQL Server Business Intelligence Development Studio from Microsoft SQL Server 2008 R2 or Microsoft SQL Server Reporting Services Report Builder 3.0.

Reports included in the system or created by user, when he uses Microsoft SQL Server 2008 R2, can be accessible through Web site using SQL Server Reporting Services (SSRS), which is a component of SQL Server.

If you use Microsoft SQL Server R2 and need a set of fields that differs from NetFlow version 5, for example, a set of fields of NetFlow version 9, in this case user can oneself create reports using tools that mention above. Those reports will be accessible through the program.

In case when you use another storage than Microsoft SQL Server 2008 R2, to perform data analysis you should use third party reporting software that allows to work with required storage type.

To generate a desired report in the **Flow collector administration** program, you should select the Reports page and in the report list select the required report name. There are the following predefined reports:

Report filename	**Brief description**
Applications pie.rdl	Provides a pie chart and a table of IP destination ports.
Equipment IPv6 traffic .rdl	Showing the distribution of traffic on the network equipment from which data are obtained if the equipment has an IP address version 6 (IPv6).
Equipment traffic .rdl	Showing the distribution of traffic on the network equipment from which data are obtained if the equipment has an IP address version 4 (IPv4).
IP traffic-compact .rdl	Displays information about network flows with the aggregation on IP addresses and ports.
IP traffic-details.rdl	Displays detail information on the network flow.
Protocols pie .rdl	Displays traffic distribution by IP protocols.
Speed per hour.rdl	Displays a graph of the average rate per hour (the amount of data per hour divided by 60 minutes).
Traffic-day.rdl	Displays the data traffic on the network per day.
Traffic-hour.rdl	Displays the data traffic on the network per day. It is recommended to

	display the data is not more than one day.

The Reports page is shown in Figure 8.

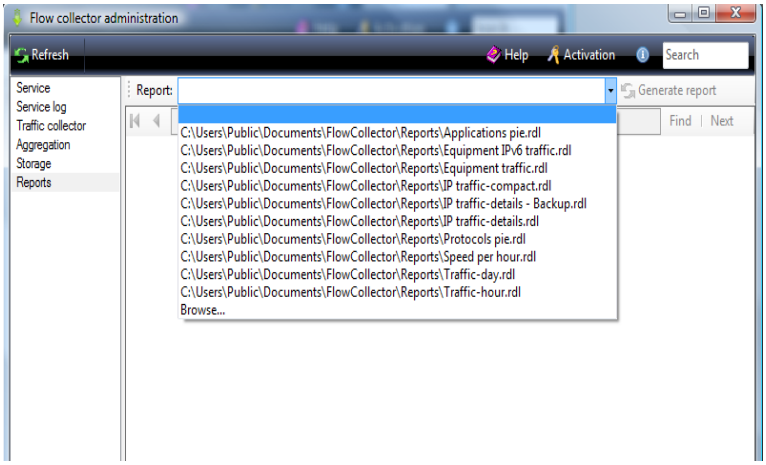

Figure 8

To generate the desired report, open the **Report** list, select the required report name, and click on the **Generate report** button. The **Report parameters** window appears. For various types of reports this window contains a different number of parameters. Parameters, for example, a time period, allow you to restrict the range of data to be processed. If necessary, the user by editing the report may change the existing parameters or add new ones. After setting the parameters the process of forming of report begins. Depending on the amount of data being processed, the parameters of the computer on which it handles, its workload with other tasks, the process of report generation can take different times.

The **Reports** page contains toolbar. The current page number of a report is displayed on the left side of the toolbar. To the right and left of this position the navigation buttons is located. When you hover over any of the buttons on the toolbar, its name is displayed. In addition to these buttons the toolbar also contains:

- "**Back to parent report**" button - It is active only when the connected reports are used that is a report contains links that allows to generate new reports;
- "**Stop report generation**" button;
- "**Refresh**" button - allows to refresh contents of report. It is relevant when you process the current data;
- "**Print report**" button;
- "**Print layout**" button - allows to display report as it will be visible in the print form;
- "**Page setup**" button - allows to change page parameters;
- "**Export report**" button - a click on this button leads to open menu that contains the next options: **Excel**, **PDF**, **Word**. Select the required option to export report in the appropriate file.;
- "**Zoom control**" list;
- "**Search text**" box and "**Find**" button - allows to find the required data in a text report;
- "**Next**" button - provides the next step of data search in the report.

Almost all reports except the graphics data also include the table data. They are located in the pages 2+ of reports.

To create, edit as well as generate reports, user can use SQL Server Reporting Services (SSRS) that are installed simultaneously with Microsoft SQL Server 2008R2. The Reporting Services Tutorials is available: http://msdn.microsoft.com/en-us/library/ms170246(v=sql.90).aspx

Unlike the Reports page of the Flow Collector, SQL Server Reporting Services allows to save a report in next files:

- XML;
- CSV;
- TIFF;
- PDF;
- MHTML;
- Excel;
- Word.

List of fields, which can be saved in the **SoftPI Flow Collector** storage, is given in the Attachment.

Attachment. List of fields

Type	Name	Description	Data Type
IPV4_DST_ADDR	IPv4 Destination address	IPv4 Destination address	uint32
FlowDateTime	Flow date and time	Date time of flow, generated by Exporter process	unixtime
BGP_IPV4_NEXT_HOP	BGP IPV4 next hop address	Next-hop router's IP in the BGP domain	uint32
BGP_IPV6_NEXT_HOP	BGP IPV6 next hop address	Next-hop router's IP in the BGP domain	ipv6
DIRECTION	Flow direction	Flow direction: 0 - ingress flow, 1 - egress flow	byte
DST_AS	Destination AS	Destination BGP autonomous system number	uint32
DST_MASK	Destination Mask	The number of contiguous bits in the destination address subnet mask i.e. the submask in slash notation	byte
DST_TOS	Destination TOS	Type of Service byte setting when exiting outgoing interface	byte
DST_VLAN	Destination VLAN	Virtual LAN identifier associated with egress interface	uint16
ENGINE_ID	Engine ID	ID number of the flow switching engine	byte
ENGINE_TYPE	Engine Type	Type of flow switching engine: RP = 0, VIP/Linecard = 1	byte
FIRST_SWITCHED	First switched	System uptime at which the first packet of this flow was switched	uint32
FLOW_ACTIVE_TIMEOUT	Flow active timeout	Timeout value (in seconds) for active flow entries in the NetFlow cache	uint16
FLOW_INACTIVE_TIMEOUT	Flow Inactive timeout	Timeout value (in seconds) for inactive flow entries in the NetFlow cache	uint16
FLOW_SAMPLER_ID	Flow sampler ID	Identifier shown in "show flow-sampler"	byte
FLOW_SAMPLER_MODE	Flow sampler mode	The type of algorithm used for sampling data: 0x02 random sampling. Use in connection with FLOW_SAMPLER_MODE	byte
FLOW_SAMPLER_RANDOM_INTERVAL	Flow sampler random interval	Packet interval at which to sample. Use in connection with FLOW_SAMPLER_MODE	uint32
FLOWS	Flows	Number of flows that were aggregated	uint32
FORWARDING_STATUS	Forwarding status	Forwarding status is encoded on 1 byte with the 2 left bits giving the status and the 6 remaining bits giving the reason code.	byte
FRAGMENT_OFFSET	Fragment offset	The fragment-offset value from fragmented IP packets	uint32
ICMP_TYPE	ICMP type	Internet Control Message Protocol (ICMP) packet type; reported as ((ICMP Type * 256) + ICMP code)	uint16
IF_DESC	Interface description	Full interface name e.g. "FastEthernet 1/0"	string
IF_NAME	Interface name	Shortened interface name e.g. "FE1/0"	string
IN_BYTES	Incoming Bytes	Incoming counter for number of bytes associated with an IP Flow.	uint64
IN_DST_MAC	Incoming destination MAC	Incoming destination MAC address	mac
IN_PERMANENT_BYTES	Incoming permanent bytes	Running byte counter for a permanent flow	uint64

IN_PERMANENT_PKTS	Incoming permanent packets	Running packet counter for a permanent flow	uint64
IN_PKTS	Incoming packets	Incoming counter for the number of packets associated with an IP Flow	uint64
IN_SRC_MAC	Incoming source MAC	Incoming source MAC address	mac
INPUT_SNMP	Input interface	Input interface index	uint32
IP_PROTOCOL_VERSION	IP protocol version	Internet Protocol Version Set to 4 for IPv4, set to 6 for IPv6. If not present in the template, then version 4 is assumed.	byte
IPV4_DST_PREFIX	IPv4 destination prefix	IPv4 destination address prefix (specific for Catalyst architecture)	uint32
IPV4_IDENT	IPv4 identification	The IP v4 identification field	uint16
IPV4_NEXT_HOP	IPv4 next hop address	IPv4 address of next-hop router	uint32
IPV4_SRC_ADDR	IPv4 source address	IPv4 source address	uint32
IPV4_SRC_PREFIX	IPv4 source prefix	IPv4 source address prefix (specific for Catalyst architecture)	uint32
IPV6_DST_ADDR	IPv6 destination address	IPv6 Destination Address	binary
IPV6_DST_MASK	IPv6 destination mask	Length of the IPv6 destination mask in contiguous bits	byte
IPV6_FLOW_LABEL	IPv6 flow label	IPv6 flow label as per RFC 2460 definition	uint32
IPV6_NEXT_HOP	IPv6 next hop address	IPv6 address of the next-hop router	ipv6
IPV6_OPTION_HEADERS	IPv6 option headers	Bit-encoded field identifying IPv6 option headers found in the flow	uint32
IPV6_SRC_ADDR	IPv6 source address	IPv6 Source Address	ipv6
IPV6_SRC_MASK	IPv6 source mask	Length of the IPv6 source mask in contiguous bits	byte
L4_DST_PORT	L4 destination port	TCP/UDP destination port number e.g. FTP, Telnet, or equivalent	uint16
L4_SRC_PORT	L4 source port	TCP/UDP source port number e.g. FTP, Telnet, or equivalent	uint16
LAST_SWITCHED	Last packet time	System uptime at which the last packet of this flow was switched	uint32
MAX_PKT_LNGTH	Maximum packet length	Maximum IP packet length on incoming packets of the flow	uint16
MAX_TTL	Maximum TTL	Maximum TTL on incoming packets of the flow	byte
MIN_PKT_LNGTH	Minimum packet length	Minimum IP packet length on incoming packets of the flow	uint16
MIN_TTL	Minimum TTL	Minimum TTL on incoming packets of the flow	byte
MPLS_LABEL_1	MPLS Label 1	MPLS label at position 1 in the stack	uint32
MPLS_LABEL_10	MPLS Label 10	MPLS label at position 10 in the stack	uint32
MPLS_LABEL_2	MPLS Label 2	MPLS label at position 2 in the stack	uint32
MPLS_LABEL_3	MPLS Label 3	MPLS label at position 3 in the stack	uint32
MPLS_LABEL_4	MPLS Label 4	MPLS label at position 4 in the stack	uint32
MPLS_LABEL_5	MPLS Label 5	MPLS label at position 5 in the stack	uint32
MPLS_LABEL_6	MPLS Label 6	MPLS label at position 6 in the stack	uint32
MPLS_LABEL_7	MPLS Label 7	MPLS label at position 7 in the stack	uint32
MPLS_LABEL_8	MPLS Label 8	MPLS label at position 8 in the stack	uint32
MPLS_LABEL_9	MPLS Label 9	MPLS label at position 9 in the stack	uint32

MPLS_TOP_LABEL_IP_ADDR	MPLS top label IP address	Forwarding Equivalent Class corresponding to the MPLS Top Label	uint32
MPLS_TOP_LABEL_TYPE	MPLS top label type	MPLS Top Label Type: 0x00 UNKNOWN 0x01 TE-MIDPT 0x02 ATOM 0x03 VPN 0x04 BGP 0x05 LDP	byte
MUL_DST_BYTES	Multicast destination bytes	IP multicast outgoing byte counter for bytes associated with the IP Flow	uint64
MUL_DST_PKTS	Multicast destination packets	IP multicast outgoing packet counter for packets associated with the IP Flow	uint64
MUL_IGMP_TYPE	Multicast IGMP type	Internet Group Management Protocol (IGMP) packet type	byte
OUT_BYTES	Outgoing bytes	Outgoing counter for the number of bytes associated with an IP Flow	uint64
OUT_DST_MAC	Outgoing destination MAC	Outgoing destination MAC address	mac
OUT_PKTS	Outgoing packets	Outgoing counter for the number of packets associated with an IP Flow.	uint64
OUT_SRC_MAC	Outgoing source MAC	Outgoing source MAC address	mac
OUTPUT_SNMP	Output interface	Output interface index	uint32
PROTOCOL	Protocol	IP protocol	byte
SAMPLER_NAME	Sampler name	Name of the flow sampler	string
SAMPLING_ALGORITHM	Sampling algorithm	The type of algorithm used for sampled NetFlow: 0x01 Deterministic Sampling ,0x02 Random Sampling	byte
SAMPLING_INTERVAL	Sampling interval	When using sampled NetFlow, the rate at which packets are sampled e.g. a value of 100 indicates that one of every 100 packets is sampled	uint32
SRC_AS	Source AS	Source BGP autonomous system number	uint32
SRC_MASK	Source mask	The number of contiguous bits in the source address subnet mask i.e. the submask in slash notation	byte
SRC_TOS	Source TOS	Type of Service byte setting when entering incoming interface	byte
SRC_VLAN	Source VLAN	Virtual LAN identifier associated with ingress interface	uint16
TCP_FLAGS	TCP flags	Cumulative of all the TCP flags seen for this flow	byte
TOTAL_BYTES_EXP	Total bytes exported	Counter for bytes for the number of bytes exported by the Observation Domain	uint64
TOTAL_FLOWS_EXP	Total flows exported	Counter for bytes for the number of flows exported by the Observation Domain	uint64
TOTAL_PKTS_EXP	Total packets exported	Counter for bytes for the number of packets exported by the Observation Domain	uint64
BgpNextAdjacentAsNumber	bgpNextAdjacentAsNumber	The autonomous system (AS) number of the first AS in the AS path to the destination IP address.	uint32
BgpPrevAdjacentAsNumber	bgpPrevAdjacentAsNumber	The autonomous system (AS) number of the last AS in the AS path from the source IP address.	uint32
ExporterIPv4Address	exporterIPv4Address	The IPv4 address used by the Exporting Process.	uint32
ExporterIPv6Address	exporterIPv6Address	The IPv6 address used by the Exporting Process.	ipv6
DroppedOctetDeltaCount	droppedOctetDeltaCount	The number of octets since the previous report (if any) in packets of this Flow dropped by packet treatment.	uint64
DroppedPacketDeltaCount	droppedPacketDeltaCount	The number of packets since the previous report (if any) of this Flow dropped by packet treatment.	uint64
DroppedOctetTotalCount	droppedOctetTotalCount	The total number of octets in packets of this Flow dropped by packet treatment since the Metering Process (re-)initialization for this Observation Point. The number of octets includes IP header(s) and IP payload.	uint64
DroppedPacketTotalCount	droppedPacketTotalCount	The number of packets of this Flow dropped by packet treatment since the Metering Process (re-)initialization for	uint64

		this Observation Point.	
FlowEndReason	flowEndReason	The reason for Flow termination.	byte
CommonPropertiesId	commonPropertiesId	An identifier of a set of common properties that is unique per Observation Domain and Transport Session.	uint64
ObservationPointId	observationPointId	An identifier of an Observation Point that is unique per Observation Domain.	uint32
IcmpTypeCodeIPv6	icmpTypeCodeIPv6	Type and Code of the IPv6 ICMP message. The combination of both values is reported as (ICMP type * 256) + ICMP code.	uint16
MplsTopLabelIPv6Address	mplsTopLabelIPv6Address	The IPv6 address of the system that the MPLS top label will cause this Flow to be forwarded to.	ipv6
LineCardId	lineCardId	An identifier of a line card that is unique per IPFIX Device hosting an Observation Point.	uint32
PortId	portId	An identifier of a line port that is unique per IPFIX Device hosting an Observation Point.	uint32
MeteringProcessId	meteringProcessId	An identifier of a Metering Process that is unique per IPFIX Device.	uint32
ExportingProcessId	exportingProcessId	An identifier of an Exporting Process that is unique per IPFIX Device.	uint32
TemplateId	templateId	An identifier of a Template that is locally unique within a combination of a Transport session and an Observation Domain. Template IDs of Data Sets are numbered from 256 to 65535.	uint16
WlanChannelId	wlanChannelId	The identifier of the 802.11 (Wi-Fi) channel used.	byte
WlanSSID	wlanSSID	The Service Set IDentifier (SSID) identifying an 802.11 (Wi-Fi) network used. According to IEEE.802-11.1999, the SSID is encoded into a string of up to 32 characters.	string
FlowId	flowId	An identifier of a Flow that is unique within an Observation Domain.	uint64
ObservationDomainId	observationDomainId	An identifier of an Observation Domain that is locally unique to an Exporting Process.	uint32
FlowStartSeconds	flowStartSeconds	The absolute timestamp of the first packet of this Flow (seconds).	uint32
FlowEndSeconds	flowEndSeconds	The absolute timestamp of the last packet of this Flow (seconds).	uint32
FlowStartMilliseconds	flowStartMilliseconds	The absolute timestamp of the first packet of this Flow (milliseconds).	uint64
FlowEndMilliseconds	flowEndMilliseconds	The absolute timestamp of the last packet of this Flow (milliseconds).	uint64
FlowStartMicroseconds	flowStartMicroseconds	The absolute timestamp of the first packet of this Flow (microseconds).	uint64
FlowEndMicroseconds	flowEndMicroseconds	The absolute timestamp of the last packet of this Flow (microseconds).	uint64
FlowStartNanoseconds	flowStartNanoseconds	The absolute timestamp of the first packet of this Flow (nanoseconds).	uint64
FlowEndNanoseconds	flowEndNanoseconds	The absolute timestamp of the last packet of this Flow (nanoseconds).	uint64
FlowStartDeltaMicroseconds	flowStartDeltaMicroseconds	The negative time offset of the first observed packet of this Flow relative to the export time specified in the IPFIX Message Header.	uint64
FlowEndDeltaMicroseconds	flowEndDeltaMicroseconds		uint64
SystemInitTimeMilliseconds	systemInitTimeMilliseconds		uint64
FlowDurationMilliseconds	flowDurationMilliseconds		uint64
FlowDurationMicroseconds	flowDurationMicroseconds		uint64
ObservedFlowTotalCount	observedFlowTotalCount		uint64
IgnoredPacketTotalCo	ignoredPacket		uint64

unt	TotalCount		
IgnoredOctetTotalCount	ignoredOctetTotalCount		uint64
NotSentFlowTotalCount	notSentFlowTotalCount		uint64
NotSentPacketTotalCount	notSentPacketTotalCount	The total number of packets dropped by Metering Process	uint64
NotSentOctetTotalCount	notSentOctetTotalCount	The total number of octets in packets in Flow Records that were generated by the Metering Process and dropped by the Metering Process or by the Exporting Process instead of being sent to the Collecting Process.	uint64
DestinationIPv6Prefix	destinationIPv6Prefix	IPv6 destination address prefix.	ipv6
SourceIPv6Prefix	sourceIPv6Prefix	IPv6 source address prefix.	ipv6
PostOctetTotalCount	postOctetTotalCount	The definition of this Information Element is identical to the definition of Information Element 'octetTotalCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the Observation Point.	uint64
PostPacketTotalCount	postPacketTotalCount	This is the same as 'packetTotalCount', but after the packet passed the Observation Point.	uint64
FlowKeyIndicator	flowKeyIndicator	This set of bit fields is used for marking the Information Elements of a Data Record that serve as Flow Key.	uint64
PostMCastPacketTotalCount	postMCastPacketTotalCount	The total number of outgoing multicast packets sent for packets of this Flow	uint64
PostMCastOctetTotalCount	postMCastOctetTotalCount	The total number of outgoing multicast octets sent for packets of this Flow	uint64
IcmpTypeIPv4	icmpTypeIPv4	Type of the IPv4 ICMP message.	uint16
IcmpCodeIPv4	icmpCodeIPv4	Code of the IPv4 ICMP message.	uint16
IcmpTypeIPv6	icmpTypeIPv6	Type of the IPv6 ICMP message.	uint16
IcmpCodeIPv6	icmpCodeIPv6	Code of the IPv6 ICMP message.	uint16
UdpSourcePort	udpSourcePort	The source port identifier in the UDP header.	uint16
UdpDestinationPort	udpDestinationPort	The destination port identifier in the UDP header.	uint16
TcpSourcePort	tcpSourcePort	The source port identifier in the TCP header.	uint16
TcpDestinationPort	tcpDestinationPort	The destination port identifier in the TCP header.	uint16
TcpSequenceNumber	tcpSequenceNumber	The sequence number in the TCP header.	uint32
TcpAcknowledgementNumber	tcpAcknowledgementNumber	The acknowledgement number in the TCP header.	uint32
TcpWindowSize	tcpWindowSize	The window field in the TCP header.	uint16
TcpUrgentPointer	tcpUrgentPointer	The urgent pointer in the TCP header.	uint16
TcpHeaderLength	tcpHeaderLength	The length of the TCP header.	byte
IpHeaderLength	ipHeaderLength	The length of the IP header. For IPv6, the value of this Information Element is 40.	byte
TotalLengthIPv4	totalLengthIPv4	The total length of the IPv4 packet.	uint16
PayloadLengthIPv6	payloadLengthIPv6	This Information Element reports the value of the Payload Length field in the IPv6 header.	uint16
IpTTL	ipTTL	For IPv4, the value of the Information Element matches the value of the Time to Live (TTL) field in the IPv4 packet header. For IPv6, the value of the Information Element matches the value of the Hop Limit field in the IPv6 packet header.	byte
NextHeaderIPv6	nextHeaderIPv6	The value of the Next Header field of the IPv6 header.	byte

MplsPayloadLength	mplsPayloadLength	The size of the MPLS packet without the label stack.	uint32
IpDiffServCodePoint	ipDiffServCodePoint	The value of a Differentiated Services Code Point (DSCP) encoded in the Differentiated Services field.	byte
IpPrecedence	ipPrecedence	The value of the IP Precedence. The IP Precedence value is encoded in the first 3 bits of the IPv4 TOS field or the IPv6 Traffic Class field, respectively.	byte
FragmentFlags	fragmentFlags	Fragmentation properties indicated by flags in the IPv4 packet header or the IPv6 Fragment header, respectively.	byte
OctetDeltaSumOfSquares	octetDeltaSumOfSquares	The sum of the squared numbers of octets per incoming packet since the previous report (if any) for this Flow at the Observation Point.	uint64
OctetTotalSumOfSquares	octetTotalSumOfSquares	The total sum of the squared numbers of octets in incoming packets for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point. The number of octets includes IP header(s) and IP payload.	uint64
MplsTopLabelTTL	mplsTopLabelTTL	The TTL field from the top MPLS label stack entry, i.e., the last label that was pushed.	byte
MplsLabelStackLength	mplsLabelStackLength	The length of the MPLS label stack in units of octets.	uint32
MplsLabelStackDepth	mplsLabelStackDepth	The number of labels in the MPLS label stack.	uint32
MplsTopLabelExp	mplsTopLabelExp	The Exp field from the top MPLS label stack entry, i.e., the last label that was pushed.	byte
IpPayloadLength	ipPayloadLength	The effective length of the IP payload.	uint32
UdpMessageLength	udpMessageLength	The value of the Length field in the UDP header.	uint16
IsMulticast	isMulticast	If the IP destination address is not a reserved multicast address, then the value of all bits of the octet (including the reserved ones) is zero.	byte
Ipv4IHL	ipv4IHL	The value of the Internet Header Length (IHL) field in the IPv4 header.	byte
Ipv4Options	ipv4Options	IPv4 options in packets of this Flow.	uint32
TcpOptions	tcpOptions	TCP options in packets of this Flow.	uint64
PaddingOctets	paddingOctets	The value of this Information Element is always a sequence of 0x00 values.	uint32
CollectorIPv4Address	collectorIPv4Address	An IPv4 address to which the Exporting Process sends Flow information.	uint32
CollectorIPv6Address	collectorIPv6Address	An IPv6 address to which the Exporting Process sends Flow information.	ipv6
ExportInterface	exportInterface	The index of the interface from which IPFIX Messages sent by the Exporting Process to a Collector leave the IPFIX Device.	uint32
ExportProtocolVersion	exportProtocolVersion	The protocol version used by the Exporting Process for sending Flow information.	byte
ExportTransportProtocol	exportTransportProtocol	The value of the protocol number used by the Exporting Process for sending Flow information.	byte
CollectorTransportPort	collectorTransportPort	The destination port identifier to which the Exporting Process sends Flow information.	uint16
ExporterTransportPort	exporterTransportPort	The source port identifier from which the Exporting Process sends Flow information.	uint16
TcpSynTotalCount	tcpSynTotalCount	The total number of packets of this Flow with TCP "Synchronize sequence numbers" (SYN) flag set.	uint64
TcpFinTotalCount	tcpFinTotalCount	The total number of packets of this Flow with TCP "No more data from sender" (FIN) flag set	uint64
TcpRstTotalCount	tcpRstTotalCount	The total number of packets of this Flow with TCP "Reset the connection" (RST) flag set.	uint64
TcpPshTotalCount	tcpPshTotalCount	The total number of packets of this Flow with TCP "Push Function" (PSH) flag set.	uint64
TcpAckTotalCount	tcpAckTotalCount	The total number of packets of this Flow with TCP "Acknowledgment field significant" (ACK) flag set.	uint64

TcpUrgTotalCount	tcpUrgTotalCount	The total number of packets of this Flow with TCP "Urgent Pointer field significant" (URG) flag set.	uint64
IpTotalLength	ipTotalLength	The total length of the IP packet.	uint64
PostMplsTopLabelExp	postMplsTopLabelExp	The definition of this Information Element is identical to the definition of Information Element 'mplsTopLabelExp', except that it reports a potentially modified value caused by a middlebox function after the packet passed the Observation Point.	byte
TcpWindowScale	tcpWindowScale	The scale of the window field in the TCP header.	uint16

Technical Support

For any technical issues with the SoftPI Flow Collector please contact Technical Support:

E-mail: support@softpiua.com
Site: http://www.softpiua.com/