



- [Site Map](#)
- [Topic Index](#)

[Home](#)[Install](#)[Migrate](#)[Start](#)[Authoring](#)[Stream](#)[SQL](#)[Test/Debug](#)[Adapters](#)[API](#)[Admin](#)[Samples](#)[Studio](#)[Reference](#)

Current Location: [Home](#) > Glossary

StreamBase Glossary

A

adapter

Code that converts third-party data to the StreamBase protocol, and vice versa. For example, StreamBase provides an adapter that converts TIBCO® Rendezvous™ messages into tuples that use the data types supported by StreamBase. The adapter also supports converting StreamBase tuples into the data types used by Rendezvous messages. You can create a configuration file that specifies the source and destination message subjects (Rendezvous) and streams (StreamBase).

Aggregate operator

Applies one or more aggregating functions to windows of incoming tuples. Windows can be defined over attribute values or numbers of tuples, and can be grouped by one or more input fields. You can define several parameters, including the size of each window, how much to advance the window after each window's computation completes, and a condition that must hold for aggregation to begin.

application

A set of processing units (input streams, operators, output streams, and other components) that may be designed in StreamBase Studio and loaded by a StreamBase Server. The units define the run-time business logic that will occur on streaming input data. The design that you create and update in StreamBase Studio is known as an *EventFlow*. An application is also comprised of configuration files that set the location of application resources and declare memory parameters. A StreamBase application may also contain user-defined client programs that interact with the StreamBase components, and user-defined plugins to implement custom functions.

application module

A collection of components that can be referenced as a single component within a StreamBase application. Developing with modules enables you to reuse the same functionality in multiple applications. Any StreamBase application can be used as an application module, as long as its schema matches and its functions are compatible with the containing application.

B

boundary crossing

The exchange of data between two or more separate software applications, to accomplish a single business task.

breakpoints

In a debugger, breakpoints are a way of letting the processing engine pause for you when a certain condition on any queue is met.

BSort operator

StreamBase Glossary

Performs an approximate "buffered" sort of its input stream. Because a complete sort is not possible over an infinite stream of data, you indicate the desired degree of approximation by specifying the buffer size. You can also specify that sorting should be performed over groups specified by input data fields.

C

canvas

The drawing surface in StreamBase Studio, to which you "drag and drop" components from the Palette.

collection

A StreamBase Studio feature that lets you collapse or expand the display of multiple components on the graphical canvas. The components that comprise a collection are not combined in any way, other than for viewing. This display feature is only intended to make it easier to view large StreamBase EventFlows. On the StreamBase Studio canvas, a collection is shown as a single box with yellow input and output ports.

container

A building block for organizing and connecting multiple StreamBase applications. There is one StreamBase application per container, and each container name serves as a handle for that application's resources. Multiple containers/applications can be run in a single StreamBase Server (**sbd**) process, and you can share selected resources between applications.

data construct

A StreamBase component that stores information used by an associated StreamBase operator. For example, Query operators are associated with Query Table data constructs.

data directory

In the StreamBase context, a `datadir` is the location for storing persistent Query Tables. The `datadir` value could come from several sources; the precedence order is:

1. If specified in the **sbd** command as its `--datadir` value).
2. The `STREAMBASE_DATA` environment variable's value.
3. The value in the `server` section of the `sbd` configuration file.
4. Finally, if not specified, or the if special value `+TEMP+` is used, a temporary directory is created on server startup and deleted on shutdown. The default temporary directory is `/tmp/sbdata/<six-random-characters>`.

data parallelism

An option for operators and modules that enables concurrent processing, where you specify the number of threads that should be instantiated. Each instance runs in a separate thread. You also specify a key field expression, which causes all tuples with the same value in the specified field to be sent to the same instance.

dynamic variable

A variable you can define for an operator that can then be used in one of that operator's expression. A dynamic variable can be updated by enqueueing tuples onto an input stream or by emitting tuples on an output stream.

data type

StreamBase has a set of primitive data types, such as int, string, and boolean. Hierarchical data types, such as tuple, are nested primitive types. Some StreamBase data types have parameters. For example, string has the parameter `length`, indicating the maximum length of the string.

E

Eclipse

An open-source, Java-based integrated development environment (IDE). StreamBase Studio is based on Eclipse.

embedded adapter

An adapter that runs in the same server process (**sbd**) that is hosting and processing your StreamBase application's requests. Embedded adapters offer less client/server communication overhead and simpler administration than some external adapters. StreamBase installs a number of embedded adapters.

Error Log view

Captures warnings and errors logged by plug-ins. Show the Error Log view by selecting Window Show View Other PDE Runtime Error Log.

event

A message flowing through a stream at a particular time. The value of an event is always a tuple. While a tuple is simply a data structure, an event corresponds to something in the real world. For example, if a blue car is observed at location (100, 100), then the tuple (object-type: car, color: blue, x: 100, y:100) might be a good representation for that event.

Events are transient, although their tuples can be stored, either temporarily in operators and data constructs, or persistently in disk-based or external tables.

EventFlow

The graphical form of a StreamBase application that is created in the StreamBase Studio EventFlow Editor. An EventFlow is saved in an XML `sbapp` file and, optionally, an associated `sblayout` file.

expression

One or more symbols (arithmetic operators, function calls, constants, field names) that may be evaluated to a single-typed value given some (optional) evaluation context in which to resolve field names and other variables.

external adapter

An adapter that is installed separately from the StreamBase kit and runs in separate processes from the `sbd` daemon.

F

field

In a schema, a field refers to a specific name-data type pair. Fields can be looked up in a schema by name, and can be used to retrieve a value from a tuple. A field name must start with an alphabetic character or an underscore. It may contain alphabetic characters, numbers, and underscores.

Filter operator

Accepts a single input stream and applies one or more predicates (tests) to the arriving tuples. A predicate returns TRUE or FALSE. If a tuple's data matches a TRUE condition in a predicate, it is moved onto a designated output stream. You can define a Filter operator so that tuples not matching any of the predicates are dropped.

Focus Selector

In the StreamBase Studio, the Focus Selector allows you to click within a scaled-down version of the application's drawing, which then centers the larger-size drawing on the main canvas.

function alias

StreamBase Glossary

In StreamBase expressions, a shortcut that you can use to call a StreamBase built-in or custom function without using `calljava()` or `callcpp()`.

G

Gather operator

Concatenates tuples that share the same key value (such as a common stock symbol) from its input streams. It accepts two or more input streams, and releases a new single tuple that can be a function of any of the input tuple fields. The release happens when the Gather operator has seen a tuple with that key on each of the input streams.

Global Repository

A subdirectory under the *streambase-directory* installation that is used by StreamBase Studio to share embedded adapters. By default, a set of embedded adapters is installed with StreamBase and stored in the Global Repository. In any instance of StreamBase Studio that uses the same installation, adapters in the Global Repository are visible in the Palette's Global Adapters drawer, and can be added to an application.

The Global Repository is created in these locations:

On Windows

```
C:\Program Files\StreamBase Systems\StreamBase.n.m\lib\adapter\
```

On Linux

```
/opt/streambase/lib/adapter/
```

H

hashing algorithm

A calculation based on a primary key field value (and optionally including secondary keys) that results in a storage scheme; on a subsequent read operation, the hashing algorithm can be applied again to quickly find the matching data. For example, StreamBase uses a hashing algorithm with its [Query Table](#), each of which is a container for shared data (and to maintain state) in a StreamBase application.

heartbeat operator

Emits timer tuples at predictable intervals, on the same stream with your data tuples. This enables you to ensure that critical downstream operations occur even if there is a lull in the input stream.

I

input stream

In StreamBase Studio, an input stream is shown on the left side of the diagram as an entry point into the application. The name and schema you assign to an input stream is used by the StreamBase Server (on UNIX, sometimes called the "StreamBase Daemon") to determine whether data from clients and external sources should be enqueued onto that stream. The StreamBase Server can listen for requests from a client source, such as an **sbc enqueue** command or **sbfedsim**. The data can also arrive via an adapter that converts third-party protocols into the StreamBase protocol. Note that tuple output produced by an operator, if then consumed as input to another operator, is also considered an input stream that was generated within the application at runtime.

intermediate stream

Tuple output produced by one operator and consumed as input to another operator.

F

J

Join operator

Pairs tuples from two input streams that are within a specified distance on their respective order fields, and that satisfy an input predicate you defined.

jsbmonitor command

Displays information, in graphical format, about the status and performance of a StreamBase application. It includes the same information as the sbmonitor command, with added dynamic graphs of CPU and memory usage.

L

latency

The time between the start and end times of processing a tuple during stream combination.

layout element

In an EventFlow sbapp source file, an element that describes the edimensions or position of a component in the EventFlow Editor canvas. Contrast with semantic element.

Lock operator

Works in conjunction with an Unlock operator to only process tuples matching a set of key fields that you specified earlier in a Lock Set. This feature allows you to place other types of operators between the starting Lock operator and the ending Unlock operator, for exclusive processing of tuples of a particular type.

Lock Set

A lock set is a collection of locks that are used together by Lock and Unlock operators. A lock set has a key schema, and a specific set of values for the key schema fields will reference exactly one lock. The purpose of a Lock and Unlock operator pair is to provide exclusive access for certain tuples to a sequence of operators. This generally comes up when dealing with operators that modify shared data structures like Query Tables.

M

Map operator

Accepts a single input stream, applies an expression to the tuple values, and generates a single output stream that contains output fields whose values were determined after StreamBase evaluated the expressions. For example, an input stream might contain an item's prices in U.S. dollars (USD). A Map operator can perform the calculation to convert the prices to Euros (EUR) by applying the current conversion rate. The prices in the output stream are in EUR currency. You can also use Map operators to drop fields within a tuple.

Materialized Window

A data construct that provides a managed view of tuples passing through an input stream. The view can be based on a fixed number of tuples, a time interval, or a field value. It can also be partitioned into multiple windows. You can then directly query the data using one or more (read-only) Query operators.

Merge operator

Combines the tuples of two input streams - which must have schemas with the same fields, field names, and data types - generating a single output stream that is ordered on the same field.

metronome operator

Emits output tuples periodically based on the system clock, enabling you to control the timing of downstream components. Unlike the Heartbeat operator, Metronome does not have an input stream, and emits no other data - just a timer tuple.

StreamBase Glossary

module parameter

A variable that can be used within an expression in a StreamBase application module. The parameter's value can be set for each module reference that uses the application.

module reference

A reference to an application module that is inserted into a StreamBase Application. The module reference is treated as a single component that can be connected to other components.

Monitor view

A view in the SB Test/Debug perspective that displays information, in text format, about the status and performance of a StreamBase application. The monitor view is used only on UNIX systems; on Windows, use the Help > Launch Monitor command to display equivalent information in graphical format.

N

node

A machine (computer) on the network.

O

operator

Code that performs pre-defined work on streaming data, such as aggregating tuples, or merging tuples, or retrieving data from a Query Table. Each operator performs the work that you specify in its Properties view.

Outline view

In StreamBase Studio, the portion of the UI that shows the entities that comprise the application. An expanded list identifies each operator, data construct, input stream and output stream.

output stream

In StreamBase Studio, an output stream is shown on the right side of the application's diagram as a named exit point. Note that tuple output produced by an operator at runtime is also considered an output stream.

P

page pool

The memory pool used to hold tuples that are flowing on streams between operators.

Palette

In StreamBase Studio, the portion of the UI containing StreamBase components that you can "drag and drop" into the EventFlow Editor.

pattern query

A query that detects differences in data across one or more input streams, using the StreamBase pattern language. You can perform a pattern query in an EventFlow by using the Pattern operator; in a StreamSQL application using the SELECT statement with a FROM PATTERN clause.

perspective

In StreamBase Studio, a perspective is a window that contains several smaller windows, arranged in a certain way to facilitate the completion of a specific task. Every perspective has one window for editing files (called an "editor") and one or more windows that support the editing of those files (called "views"). StreamBase Studio has two perspectives: the SB Authoring perspective and the SB Test/Debug perspective.

predicate

M

StreamBase Glossary

In a StreamBase Filter or Join operator, a Boolean expression that evaluates to TRUE or FALSE. Example: `TRADE_ID < 7000`. When used in a Filter operator, if the predicate returns TRUE, the tuple is sent out on the output port associated with the predicate. If the predicate returns FALSE, the next predicate (if any) is evaluated. If no predicates remain, the tuple is dropped from the streaming data.

Properties tab

In StreamBase Studio, the portion of the UI that shows the parameter settings for the currently selected operator, data construct, or stream. In a Properties view, you can click Edit... to enter or modify the operator's properties.

Plug-in operator

Allows you to use additional types of operators supplied by StreamBase Systems or our partners.

Q

Query operator

An operator that can write to (populate) or read from a Query Table. This allows you to maintain state and share data across the StreamBase application.

Query Table

A container for shared data and to maintain state in a StreamBase application. At runtime, a Query Table can have its values read from or updated by one or more Query operators. Query Tables can be declared to reside in-memory, or on-disk. In the case of memory-resident Query Tables, any data stored in the table at runtime is not saved after the StreamBase Server is shut down. In the case of disk-based Query Tables (available only in StreamBase Enterprise Edition), the data stored in the tables can persist between StreamBase Server sessions, provided that you defined a data directory.

R

row

A row is an entry in a relational table within a data construct (such as a Query Table or JDBC Table) or a Materialized window. The value of a row is always a tuple.

S

sbadmin

The command to invoke the StreamBase Administration Client (sbadmin).

sbapp

XML source file for an EventFlow application.

sbc

The command to invoke the StreamBase Client (sbc).

sbdb

The command to invoke the StreamBase Server (sbdb).

sbdb

The command to invoke the StreamBase Debugger (sbdb).

sbdb.sbconf

The skeleton configuration file for the StreamBase Server. There is one sbdb.sbconf file per StreamBase installation on a node (a physical machine). By default sbdb.sbconf resides in `/opt/streambase/etc`.

sbfeedsim

The command to invoke the StreamBase Feed Simulator (sbfeedsim).

sbstudio

P

StreamBase Glossary

- The command to invoke StreamBase Studio (**sbstudio**).
- sbuseradmin**
The command to invoke the StreamBase Administration Client (**sbadmin**).
- sb-config**
The command to invoke the StreamBase Configuration Tool (**sb-config**).
- schema**
A description of valid fields in a tuple, defining acceptable names and data types.
- semantic element**
In an EventFlow sbapp source file, an element that defines an application component or functionality. Contrast with layout element.
- Split operator**
The Split operator allows you to explicitly indicate the order in which the outputs should be processed (to completion) by each set of downstream components, in cases where you have a single input port and multiple output ports.
- stream**
An ordered and potentially infinite collection of tuples that share the same schema and arrive at the same processing input. A stream has one producer, which is either an operator in the application or an external data source. A stream may also have zero or more consumers, which may be operators or output streams.
- StreamBase Administration Client (**sbadmin**)**
A tool that submit commands to perform administration tasks. Optionally, you can limit the use of **sbadmin** commands to designated user accounts, by enabling an authentication feature. The **sbadmin** commands include **pause**, **resume**, and **shutdown**.
- StreamBase Client (**sbc**)**
A command-line tool that sends requests to the StreamBase Server process (**sbd**). Any StreamBase user can submit **sbc** commands, which do not require authentication. The **sbc** commands include **describe**, **dequeue**, **enqueue**, **list**, **status**, and **typecheck**
- StreamBase Configuration Tool (**sb-config**)**
On UNIX, lets you determine the compiler and linker flags that should be used when compiling and linking client programs against the StreamBase libraries, for C/C++ or Java.
- StreamBase Debugger (**sbdb**)**
A debugging tool to pause a running StreamBase application, and then step through the processing to understand how and when individual operators are handling tuples. The **sbdb** debugger includes options to pause, dump, edit, list, get status, step, and continue.
- StreamBase Feed Simulator (**sbfeedsim**)**
A feature that connects to a running StreamBase application and submits data to it. This feature can help you quickly test the validity and performance of your application.
- StreamBase Processing Engine**
The core StreamBase Server.
- StreamBase Server (**sbd**)**
A server process that listens for, and acts upon, client requests (commands and data) in a StreamBase application. Also known as the StreamBase Daemon. This process is the primary server for StreamBase, accepting the incoming streaming data, managing the execution queues, and performing the real-time processing work as defined in the operators that comprise your application.
- StreamBase Studio (**sbstudio**)**
A graphical tool to create and test StreamBase applications. Started on Windows by clicking the StreamBase Studio desktop icon, or on Linux with the **sbstudio** command. You can "drag and drop" different types of StreamBase entities, such as an Input Stream or a Join operator, onto a drawing canvas. Once placed on the canvas you can draw arrows between related items, indicating that data will flow from one operator to the other, when one or more conditions are met. StreamBase Studio provides a quick and easy way to define or change each operator's properties. Behind the scenes, the

StreamBase Glossary

Studio creates the corresponding XML definitions for your application.

StreamBase User Administration Tool (**sbuseradmin**)

A tool to add or remove user names and passwords from a file-based authentication list. If StreamBase authentication has been enabled in the server's `sbd.sbconf` file, only those users who correctly enter a valid user name and password combination may execute an **sbadmin** command. The user names and passwords entered by the **sbuseradmin** command are saved in the **sbpasswd** file. By default, the file resides in `/opt/streambase/etc`.

StreamSQL

A SQL-like text language for coding StreamBase applications that are functionally equivalent to graphical EventFlow applications. StreamBase Studio includes a StreamSQL editor that facilitates writing StreamSQL applications.

T

tuple

A set of data, associated with an event message, that is allowed to flow through a stream. A tuple is composed of one or more fields; each field is a name-value pair. Each tuple must conform to the stream's schema, which specifies permissible names and data types.

typecheck

A determination that tuples output by one operator have the schema required to be processed by the operator(s) to which they are routed. Typechecking also includes validation of the operator's properties, including the syntax of the XML elements that define the properties. This validation is done because a user may have edited the `.sbapp` file's XML directly.

type coercion

Type coercion can occur when a function expects an argument of one data type but is given a value of a different data type. The given value is automatically converted to the expected type. Coercion occurs only between specific types. For example, all types can be coerced to string, and some numeric types can be coerced into one another. The value `null` can be coerced into any type.

type conversion

Type conversion is an explicit action by a user to convert from one type to another, using a type-constructor (in the API) or a StreamBase function (in expressions). For example, the expression

```
string(12)
```

will create a string with the value `12`. Type conversion is legal for many more pairs of types.

type supersetting

Type supersetting occurs when the system is presented with two types, and comes up with a common supertype. This occurs in the Union operator, for example.

U

UDA

User Defined Aggregate, an aggregate function that you can define and use in an Aggregate operator.

UDF

User Defined Function, a function that you can define and use in any expression in StreamBase, such as in a Map or Filter operator.

Union operator

An operator that merges two or more input streams of tuples - which must have the same schema - into a single stream in the order they arrived.

Unlock operator

StreamBase Glossary

Works in conjunction with a Lock operator to only process tuples matching a set of key fields that you specified earlier in a Lock Set. This feature allows you to place other types of operators between the starting Lock operator and the ending Unlock operator, for exclusive processing of tuples of a particular type.

V

view

In StreamBase Studio, a view is one of the smaller windows within a perspective that facilitates the completion of a specific task.

W

wildcard rule

In StreamSQL, a target list entry in a SELECT statement that uses one or more asterisk (*) wildcards instead of explicit return values. Wildcard rules are expanded to a list of normal expressions.

window

A sampling of tuples from an input stream that provides a tailored view of the data. Several operators, including Aggregate and Join, perform their operations on windows, and provide parameters that you can use to control the behavior of windows. Windows can be defined over attribute values or numbers of tuples, and can be grouped by one or more input fields. To determine the rate at which windows advance over a data stream, they can be bounded by a maximum number of tuples, by time, or by a range of values in a field.

workspace

An area of the file system containing projects, application files, and metadata used to manage StreamBase projects and sessions. You can have multiple workspaces, each focusing on a different area of development.

- Copyright © 2004-2009 StreamBase Systems, Inc. All Rights Reserved.
- Contact Us