

## How the SRC-VTA Works

---

The SRC-VTA manages subscriber accounts using a rule-driven event-processing system that can prioritize the actions taken for certain conditions. The SRC-VTA is triggered by events, such as the logging in of subscribers, the use of network services, or the changing of account balances. These events can cause actions, such as updating account balances, starting or stopping network services, or running scripts to perform external actions.

A VTA processes external events based on its configuration. A VTA configuration is made up of:

- Event handlers
- Actions
- Processors

## Events

Each VTA event corresponds to one subscriber and contains some attributes. The SRC-VTA supports the following types of events:

- Service- and subscriber-tracking events from the SAE; for example, start or stop tracking events.
- Account update events triggered by updating database accounts.
- Callback events triggered by a method call.

## Event Attributes

Each event carries attributes. Table 1 on page 1 describes the types of attributes that are available for each type of event.

**Table 1: Event Attributes**

Event Type	Available Attributes
Service and subscriber tracking events from the SAE	<ul style="list-style-type: none"><li>■ Plug-in attributes, such as PA_SERVICE_NAME or PA_LOGIN_NAME, associated with an SAE plug-in event. For a list of SAE plug-in events, see the SAE CORBA plug-in SPI online documentation on the Juniper Networks Web site at <a href="http://www.juniper.net/techpubs/software/management/sdx/api-index.html">http://www.juniper.net/techpubs/software/management/sdx/api-index.html</a> or in the <b>SDK+AppSupport+Demos+Samples.tar.gz</b> file on the Juniper Web site at <a href="https://www.juniper.net/support/csc/swdist-erx/src.html">https://www.juniper.net/support/csc/swdist-erx/src.html</a>.</li><li>■ currentTime attribute—Time since January 1, 1970 UTC when the SRC-VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807.</li><li>■ subscriberId—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.</li></ul>

**Table 1: Event Attributes** *(continued)*

Event Type	Available Attributes
Account update events	<ul style="list-style-type: none"> <li>■ <code>old_status_&lt;accountname&gt;</code> —Returns the old status of the account.</li> <li>■ <code>new_status_&lt;accountname&gt;</code> —Returns the new status of the specified account.</li> <li>■ <code>old_lastUpdateTime_&lt;accountname&gt;</code> —Returns the old last update time of the account.</li> <li>■ <code>new_lastUpdateTime_&lt;accountname&gt;</code> —Returns the new last update time of the account.</li> <li>■ <code>old_balance_&lt;accountname&gt;</code> —Returns the old balance of the account.</li> <li>■ <code>new_balance_&lt;accountname&gt;</code> —Returns the new balance of the specified account</li> <li>■ The <code>currentTime</code> attribute, which is the time since January 1, 1970 UTC, when the SRC-VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807.</li> <li>■ <code>subscriberId</code>—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The <code>subscriberId</code> event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.</li> </ul>
Callback events	<ul style="list-style-type: none"> <li>■ <code>callId</code>—Includes all parameters provided with the callback.</li> <li>■ <code>currentTime</code>—Time since January 1, 1970 UTC, when the SRC-VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807.</li> <li>■ <code>subscriberId</code>—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The <code>subscriberId</code> event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.</li> </ul>

## Event Handlers

An event handler defines how the SRC-VTA processes an event. VTA event handlers consist of:

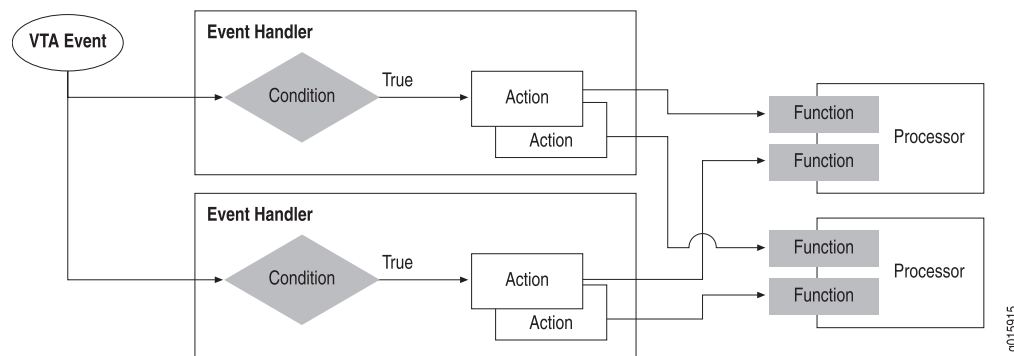
- **Type of event**—Tracking, update, or callback event; for example, a subscriber start-tracking event or a service start-tracking event.
- **Priority**—Priority at which event handlers are evaluated and executed. Event handlers are evaluated and executed from high to low priority.
- **Condition**—Condition that the event handler evaluates to determine whether the event handler should handle the event.
- **Actions**—List of actions to be performed by the event handler.

You can set up multiple event handlers to process events. For example, the first event handler could retrieve the balance for a quota account, and the next event handler could refill the quota account, depending on whether the condition of the second event handler is met.

Figure 1 on page 3 shows the event handler model. The SRC-VTA processes an event as follows:

1. The event handler with the highest priority receives the event, determines whether the event's type is the same as the event type of the event handler, and determines whether the event satisfies the condition of the event handler.
2. If the condition is met, the SRC-VTA performs the corresponding actions based on the event attributes. An action invokes a function and provides the parameters required by that function to the processor.
3. When an event handler finishes processing an event, the next applicable event handler according to the priority of the event handler processes the event.

**Figure 1: VTA Event Handler Model**



## Actions

You can specify actions that the SRC-VTA takes in response to an event; for example, updating an account balance, starting a service, or stopping a service. The actions are performed if the event is of the specified type and matches the specified condition. An action is modeled as a call to a function.

An action can update event attributes or add new attributes to an event for subsequent processing of the same event by another event handler.

An action configuration includes:

- Function—Function that the action invokes
- Parameter—Parameters and corresponding values to be passed to the function

## Processors

Processors receive and process events. The SRC-VTA has four processors:

- Database engine processor—Acts as a proxy to a database.
- Mail processor—Sends e-mail notifications when certain events occur.
- SAE proxy processor—Acts as a proxy to the SAE.
- Script runner processor—Runs external scripts or JavaScript programs.

## **Database Engine Processor**

The database engine processor is a proxy to a database. You can use the database engine processor to:

- Calculate the use of network resources for a service.
- Calculate the interim accounting interval for each service based on a subscriber's remaining resources and use of the service.
- Update VTA accounts with a JavaScript program.
- Terminate a VTA session. This feature is usually used at the end of a billing period so that you can finish collecting data for the current billing period and start a new VTA session for the new billing period.

## **Mail Processor**

You use the mail processor to set up the SRC-VTA to send e-mail notifications to subscribers when certain events occur.

## **SAE Proxy Processor**

You can use the SAE proxy processor to:

- Set an interim interval for a service.
- Set a service session timeout for a subscription.
- Set a session timeout for a subscriber.
- Start a subscription to a service. You can specify the parameter substitutions to use when the service is started.
- Stop a subscription to a service. You can include a reason for the subscription's being stopped. When the service is stopped, the reason is sent to the billing system so it can differentiate between service stops.

The SAE proxy processor is a proxy to the SAE external interface that resolves the subscriber interface based on the event types to which functions are applied.

- If a function is applied to SAE subscriber-tracking or service-tracking events, the processor finds the SAE reference in the event message.

There is an exception if the `CurrentSubscriberOnly` parameter is set to false. In this case, the function finds subscribers in all SAEs with the NIC.

- If a function is applied to other events, the processor uses subscriber's ID in the event as the key for the NIC to find the SAE reference.

## **Script Runner Processor**

The Script runner processor can invoke external executable scripts or JavaScripts. We recommend using JavaScript, where possible, for better performance.

- External scripts are executable programs, such as shell scripts, that are available on the SRC-VTA's host. Each external script can perform a task and return a value. If the script returns a value, the value can be added to the current event as an event attribute.
- JavaScript programs are used to process attributes of a VTA event. For example, it can convert a VTA event attribute in a timestamp to a date string and add it to the event as a new attribute. The attribute can then be used for subsequent actions, such as sending an e-mail notification to the subscriber. The JavaScript program can refer to any attributes of the event being processed, and it must return a value.

**Related Topics**

- Overview of the SRC-VTA
- SRC-VTA Operation
- Installing the SRC-VTA and Running the Configuration Script
- Configuring Events
- Configuring Event Handlers
- SRC-VTA Architecture and Connections to SRC Components

