Health Monitor Events

SCA application integrates with the Global Diagnostic app to identify the occurrence of specific events during the application flows, classify them and send appropriate notification to Verifone event server. The payment application provides the required data along with the event message to help troubleshoot or acquire further understanding regarding the event.

A separate application called GDA has been developed to collect health monitor parameters as events from the payment application and transmit them to the Event service on Verifone server. The application independently monitors certain system level parameters such as signal strength, network availability, battery level etc.

There are 21 events currently supported by the Health Monitor functionality which are classified as Critical and Non-Critical events. **Critical events** may cause the payment application to fail initialization therefore rendering it useless or when the device does not have connectivity to the host causing SAF build up. The Critical events are posted instantly once it is identified and allow the consumer to act on it immediately. **Non-Critical events** have a cadence of once every 24 hours. This typically occurs after the 24-hour reboot.

These 21 events are posted into Verifone event server to allow customers to register webhooks to notify directly during the occurrence of the events. This allows customers to maintain the required dashboard for monitoring and acting on such events. Alternatively, customers can choose to sign up for sending emails and pulling additional diagnostic data from the devices during the events.

The health monitor functionality brings enables several internal and customer teams for Troubleshooting and diagnosing issues based on captured events. It allows customers to gain insights into trending issues in production and take proactive measures to arrest them.

Events Format

Webhook Format:

Following is the generic format for the events that a subscriber will receive as a webhook notification.

Content Type: application/json

```
{
"objectType": "DeviceEvent",
"eventId": "d94a63e5-3018-4265-a0da-36783d6661ad",
"eventDateTime": "2025-03-26T05:33:27.001Z",
"recordId": "0",
"itemId": "0",
"entityUid": "76e84daa-c954-4c6a-8f7f-09758c078669",
"eoEntityUid": "76e84daa-c954-4c6a-8f7f-09758c078669",
"source": "device",
"received": "2025-03-26T05:33:28.500035811Z",
"eventType": "",
"content": {
},
"deviceType": "",
"serialNumber": ""
}
```

Where,

Fields	Description

objectType:	Type: string Value: DeviceEvent Type of Event or Object. Events from Health monitor and SCA would send value of DeviceEvent.
eventId:	Type: string <uuid> The internally assigned identifier for the event.
eventDateTime:	Type: string <date-time> The timestamp when the event occurred. This should be in local time with an offset to UTC specified to ensure this can be converted between time zones.
recordId:	Type: string
itemId:	Type: string
entityUid:	Type: string <uuid> The Verifone allocated unique id for the entity this webhook/event is for.
eoEntityUid:	Type: string <uuid> The internally assigned identifier for the estate owner/customer for the event.
Source:	Type: string Value: device This internally assigned value indicates that the source of the event is from the Device. All health monitor events will have a source assigned to Device.
Received:	Type: string <date-time> (eventDateTime) This timestamp indicates when the event was received to the Verifone server Database.

eventType:	Type: string The value indicates the type of Event. The various possible event Types is provided in the next section.
content:	Type: object The associated event data. This is optional and if specified will vary according to the event type.
deviceType:	Type: string This value indicates the Device model from which the event has originated. Different Values supported - M400, M440, M424, P400, V400, E285, E280, P200
serialNumber:	Type: string. The device hardware serial number.

The above table describes the basic Event Format which all events would adhere to. The following is the **Supported List of Events**:

- Device Registration Failure
- Encryption Setup Failure
- Test Sale Failure
- Canadian Debit MAC Key Setup Failure
- Application Abrupt Restarts or Abrupt Terminal Reboots
- Host Connection Status
- Multiple SAF Retry
- Application Crash or Hang
- Payment Ready/ Not Ready State
- Host Response Timeout
- POS Disconnection
- Last Transaction Queries
- SAF Details
- Duplicate Transactions
- Network Down
- Poor Signal Strength
- Low Battery
- Host Declined Transactions
- Transaction Performance Time
- Fallback Transactions
- Card Not Supported

Note

Refer to SCA Payment Application Health Monitor User Guide for more information on the events and how this feature can be leveraged.

Events Throttling

Global diagnostic App throttling logic to reduce the possibility of blocking up the network, due to all devices in customer estate posting the diagnostic details at the same time after the 24-hour reboot occurs.

This throttling logic ensures different devices post the non-critical events at different intervals and it is calculated based on the Serial number. For example, if the serial number of the device is 321-456-789 and throttling window is set to 600 seconds (10 minutes), then the throttling time is calculated as 321456789 Mod (modulo) 600 which sets the throttling time to 189 seconds. The events will be posted 189 seconds after the 24-hour reboot window for that particular device.

Configuration

This feature can be enabled by setting HEALTHMONITOR parameter (Application Parameters). This parameter allows sending SCA application diagnostics events messages to GDA and forwarded to the host. The default setting for this parameter is disabled.

HEALTHMONITOR_EXCLCMDS parameter (Application Parameters) is also added to set the list of commands which can be excluded from the diagnostics Event Types **ApplicationException** and **AppAbruptRestarts**. The list of commands should be presented as separated by pipe (|).