



# Peripheral Gateway Processes

---

- [Overview, on page 1](#)
- [Peripheral Interface Manager, on page 1](#)
- [Message Delivery Service, on page 1](#)
- [Open Peripheral Controller, on page 2](#)
- [PG Agent, on page 2](#)
- [Computer Supported Telephony Application Message Example, on page 2](#)
- [Two Models of Reporting \(Webex CCE\), on page 4](#)

## Overview

Four processes on the Peripheral Gateway are critical to reporting: the Peripheral Interface Manager (PIM), the Message Delivery System (MDS), the Open Peripheral Controller (OPC), and the Peripheral Gateway Agent process (PG Agent).

## Peripheral Interface Manager

The Peripheral Interface Manager (PIM) manages communication between the PG and the peripherals themselves (ACDs, IVRs). The PIM's main function is to convert peripheral-specific events and requests to a Webex CCE-compatible peripheral data stream.

The PIM supplies the Open Peripheral Controller (OPC) with Computer-Supported Telephony Application (CSTA) call event reporting messages. These messages form the basis of real-time monitoring and historical reporting. The OPC process receives the CSTA messages from the PIM and uses them to construct the actual real-time and historical routing and reporting data.

## Message Delivery Service

The Message Delivery Service (MDS) manages all data flow between Webex CCE processes within the PG. The MDS notifies connected processes of errors detected during a data flow request. In addition, it plays a key role in keeping duplexed components (such as Loggers) synchronized.

# Open Peripheral Controller

The Open Peripheral Controller (OPC) is the process that takes real-time data and events from the PIM and presents these data to the CallRouter. The OPC process forms the database objects the CallRouter needs to route calls and to monitor real-time activity on the peripheral. These include call objects, agent objects, Service objects, Peripheral device objects, routing objects, and skill groups.

To interface with the PIM, OPC uses the *OPC Interface*. The OPC Interface provides a standard communication interface between OPC and the various types of PIMs.

The OPC process outputs the data it receives from the PIM in the form of OPC Interface (OPCI) messages, which OPC uses to track the state transition of monitored calls and agents. The OPCI messages are based on European Computer Manufacturers Association (ECMA) Standard Protocol for Computer-Supported Telephony Application (CSTA). They also include additional components and interfaces to support real time data feeds or other call control interfaces needed for an ACD.

## Open Peripheral Interface Data Elements

To interface with the Central Controller Agent, OPC uses the Open Peripheral Interface (OPI).

The OPI defines the objects that control the flow of OPCI messages from OPC to the CallRouter. Each table in the Central Database has a set of fields that the CallRouter uses to make its routing decisions. OPI defines tags for each of those fields.

As elements change based on events and updates from the ACD, OPC informs the CallRouter of the changed values based on table type, tag, and value. OPC sends to the Router only those data elements that have changed in value. Types of OPI data elements reported to the CallRouter are Now, Half, and Today.

### Related Topics

[Real-time data](#)

## PG Agent

The PG Agent process is responsible for controlling the flow of OPI messages from OPC to the CallRouter. It manages all message traffic between the Peripheral Gateway and the Agent Process on the CallRouter, which is called the Central Controller Agent (CC Agent). The protocol used between the two agent processes is the Device Management Protocol (DMP).

## Computer Supported Telephony Application Message Example

To illustrate how Computer Supported Telephony Application (CSTA) messages from the PIM are translated into OPI data elements, it helps to examine one CSTA message: *CSTAEstablished*.

Several OPC state transitions occur when OPC receives this message. The *CSTAEstablished* event indicates that a device (agent, trunk, or voice port) answered a call.

When OPC receives this event, the following OPC state transitions take place:

- **If the call was Queued**, several database elements and call objects are changed:

- The count for CallsQNow is reduced by one (-1).

CallsQNow is a database element for services and routes that tracks the number of calls currently in queue at the peripheral.

- The call object used to track the CallsQNow and CallQNowTime data elements is removed from the Call Queued object for the service and route associated with the call.

CallsQNowTime is a database element that records the time in seconds that all calls currently in queue to the service or route have spent in the queue.

- The CallsLeftQTo5 data element for the service or route associated with the call increases by one (+1).

CallsLeftQ is a database element that provides the total number of calls to the service or route that were removed from queue during the current five-minute interval. CallsLeftQ is also used to calculate expected delay.

- LocalQTime is written in the Termination\_Call\_Detail table.

LocalQTime is the time in seconds that the call was in the local queue at the peripheral. The Termination\_Call\_Detail record contains information about how each call was handled at a peripheral. It is generated for each call that arrives at a peripheral (provided the proper monitoring is enabled for the peripheral).

- **If there is a Call Alert event**, the amount of time the call spent ringing is added to the Call object for RingTime in the Termination\_Call\_Detail record.

RingTime is the number of seconds that the call spent ringing at the agent's teleset before being answered.

- **If the answering device is an agent**, the following data elements and call objects are changed:

- The AgentsTalking data element for the service or route associated with the call is increased by one (+1).

AgentsTalking is a service and route database element that provides a count of the number of service agents currently in one of several talking states.

- The call is associated with the agent and the agent is placed in the TalkingIn state on behalf of the call. This increases by one (+1) the count for TalkingIn for the skill group associated with the call the agent is handling.

TalkingIn is a database element for skill groups that provides a count for the number of agents in the skill group currently talking on inbound calls.

- The parameters used to calculate the database element AvgSpeedAnswer are modified.

AvgSpeedAnswer is a service and route data element. It provides the average AnswerWaitTime for all calls to the service or route (that is, the average time that all calls to the service or route had to wait before being answered). The calculation for AvgSpeedAnswer is  $\text{AnswerWaitTime} / \text{CallsAnswered}$ .

- The CallsAnsweredHalf (in the real-time database tables) and CallAnsweredTo5 (in the five-minute tables), are increased by one (+1).

- The AnswerWaitTime for the call is added and written to the database.

AnswerWaitTime is the elapsed time from when the call was offered at the peripheral to when it was answered. It includes any RingTime, LocalQTime, and DelayTime (all from the Termination\_Call\_Detail records) associated with calls.

- RingTime, LocalQTime, and DelayTime are added to AnswerWaitTimeTo5.
- TalkTime for the call begins to be monitored.

TalkTime is a service completed call time data element. It is populated with TalkTime and HoldTime from the Termination\_Call\_Detail record for the call. The value is not updated in the database until any after-call work associated with the call is completed.

## Two Models of Reporting (Webex CCE)

The PIM is responsible for general monitoring functions that include accessing data on the peripheral regarding agent groups, service, routes, trunk groups, and agents.

The level of data provided by the PIM is determined by the types of CTI links available on the peripheral. The PIM can retrieve ACD statistics by using an event-based CTI feed, an aggregate data CTI feed, or both. In general, an event-based CTI feed provides more data and capabilities than an aggregate data CTI feed.

### Event-Based Reporting

An event-based PIM (for example, the Aspect Event Link PIM) connects to a CTI link that provides call events and agent state events.

Event-based PIMs base their data on agent and call state transitions reported from the ACD. These PIMs convert CTI events received from the switch to CSTA-based OPC API messages, which can then be forwarded to OPC. The OPC constructs the routing and monitoring data from these events.

### Aggregate-Based Reporting

Some aggregate-data PIMs connect to CTI links that provide aggregate skill group, service, and route statistics. The aggregate-data PIM works by polling the ACD for certain data supported over the switch vendor's CTI link. The aggregate PIM reports to OPC those data components that are required to build the OPI data elements.

When the PIM detects a change, it updates OPC, which informs the CallRouter of the changed values. ACD-specific data is supported by a passed-through API defined in the OPC interface. OPC sends the data to the CallRouter to be stored in the Central Database. Pass-through data requires that the database define the table format of the records.