This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Genesys Pulse Deployment Guide

Genesys Pulse 8.5.0

1/3/2022

# Table of Contents

# Pulse Deployment Guide

Pulse is a Genesys Administrator Extension (GAX) plug-in application that enables at-a-glance views of real-time contact center statistics within the GAX graphical user interface. Pulse uses widgets to display user-defined List, Donut, Key Performance Indicator (KPI), or Grid charts of statistics for objects.

> **Important**
>
> This Deployment Guide provides instructions for a new installation of Pulse. To migrate from an earlier version of Pulse, start with the *Pulse 8.5 Deployment Procedure*.

### About Pulse

Find overview information about architecture, components, and other concepts for Pulse.

Introducing Pulse

Architecture

Framework Components

### Deploying Pulse

Plan your environment and install deploy Pulse.

Deploying Pulse and Pulse Collector

Configurations Options

Application Files

### Starting and Stopping

Start or stop the applications:

Pulse

Pulse Collector

### Change History

Find information about what has changed in this release of the content.

Change History

# Introducing Pulse

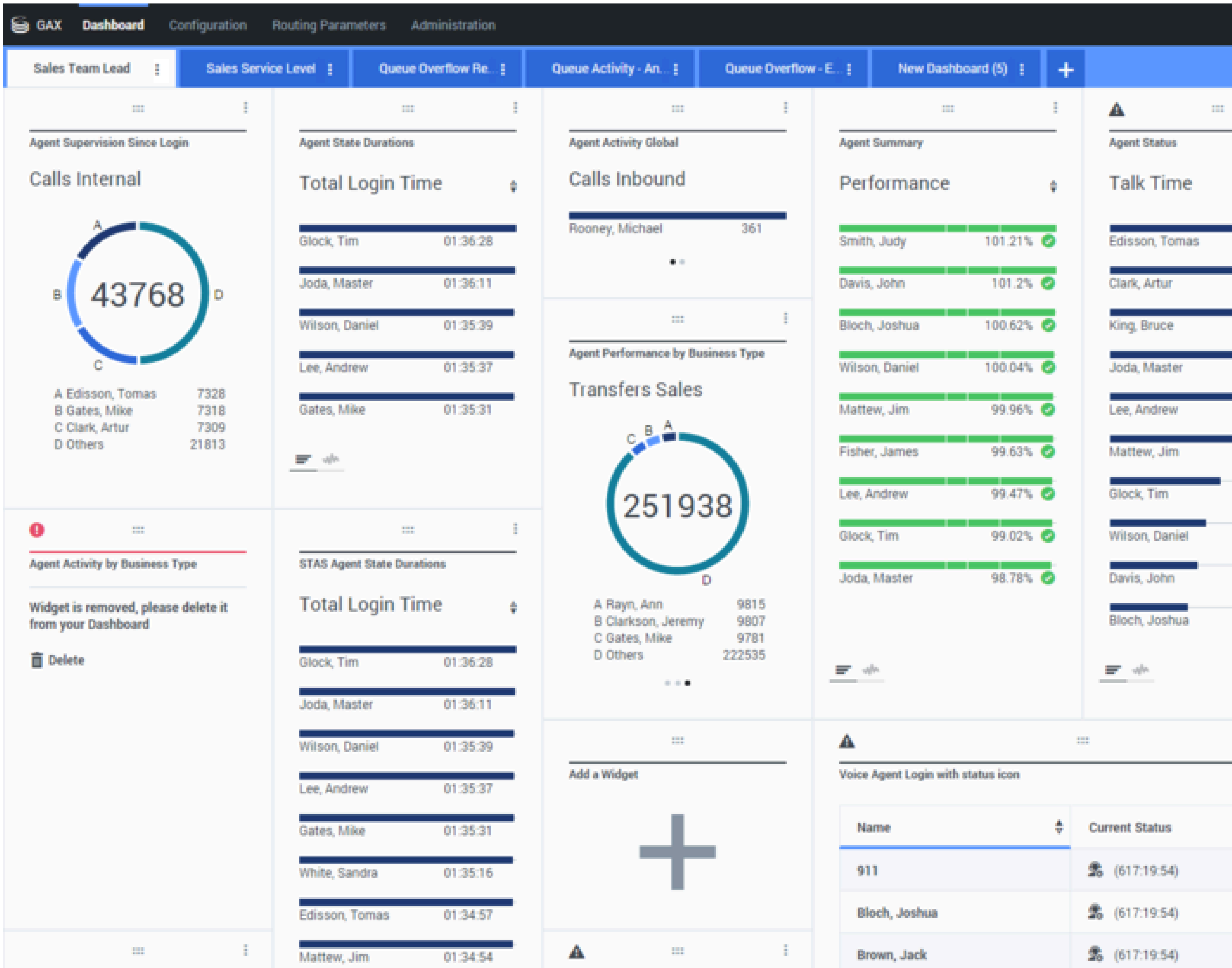
Genesys Pulse is a widget-driven, graphical user application, which is accessible from a web browser as a Genesys Administrator Extension (GAX) plug-in application. Using a direct communication link to a real-time metrics engine, Stat Server, Genesys Pulse enables at-a-glance views of real-time contact center statistics within the GAX user interface.

The tabs below provide a quick visual representation of some of the Pulse functionality and a list of the Genesys-provided templates. For more details, see the Pulse User's Guide.

## Dashboard

## Pulse Dashboard Examples

1. Examples of a Sales Team Lead dashboard



Agent Supervision Since Login

Calls Internal

43768

| | |
|---|---|
| A Edisson, Tomas | 7328 |
| B Gates, Mike | 7318 |
| C Clark, Artur | 7309 |
| D Others | 21813 |

Agent State Durations

Total Login Time

| | |
|---|---|
| Glock, Tim | 01:36:28 |
| Joda, Master | 01:36:11 |
| Wilson, Daniel | 01:35:39 |
| Lee, Andrew | 01:35:37 |
| Gates, Mike | 01:35:31 |

Agent Activity Global

Calls Inbound

| | |
|---|---|
| Rooney, Michael | 361 |

Agent Summary

Performance

| | |
|---|---|
| Smith, Judy | 101.21% ✓ |
| Davis, John | 101.2% ✓ |
| Bloch, Joshua | 100.62% ✓ |
| Wilson, Daniel | 100.04% ✓ |
| Mattew, Jim | 99.96% ✓ |
| Fisher, James | 99.63% ✓ |
| Lee, Andrew | 99.47% ✓ |
| Glock, Tim | 99.02% ✓ |
| Joda, Master | 98.78% ✓ |

Agent Status

Talk Time

Edisson, Tomas
Clark, Artur
King, Bruce
Joda, Master
Lee, Andrew
Mattew, Jim
Glock, Tim
Wilson, Daniel
Davis, John
Bloch, Joshua

Agent Activity by Business Type

Widget is removed, please delete it from your Dashboard

Delete

STAS Agent State Durations

Total Login Time

| | |
|---|---|
| Glock, Tim | 01:36:28 |
| Joda, Master | 01:36:11 |
| Wilson, Daniel | 01:35:39 |
| Lee, Andrew | 01:35:37 |
| Gates, Mike | 01:35:31 |
| White, Sandra | 01:35:16 |
| Edisson, Tomas | 01:34:57 |
| Mattew, Jim | 01:34:54 |

Agent Performance by Business Type

Transfers Sales

251938

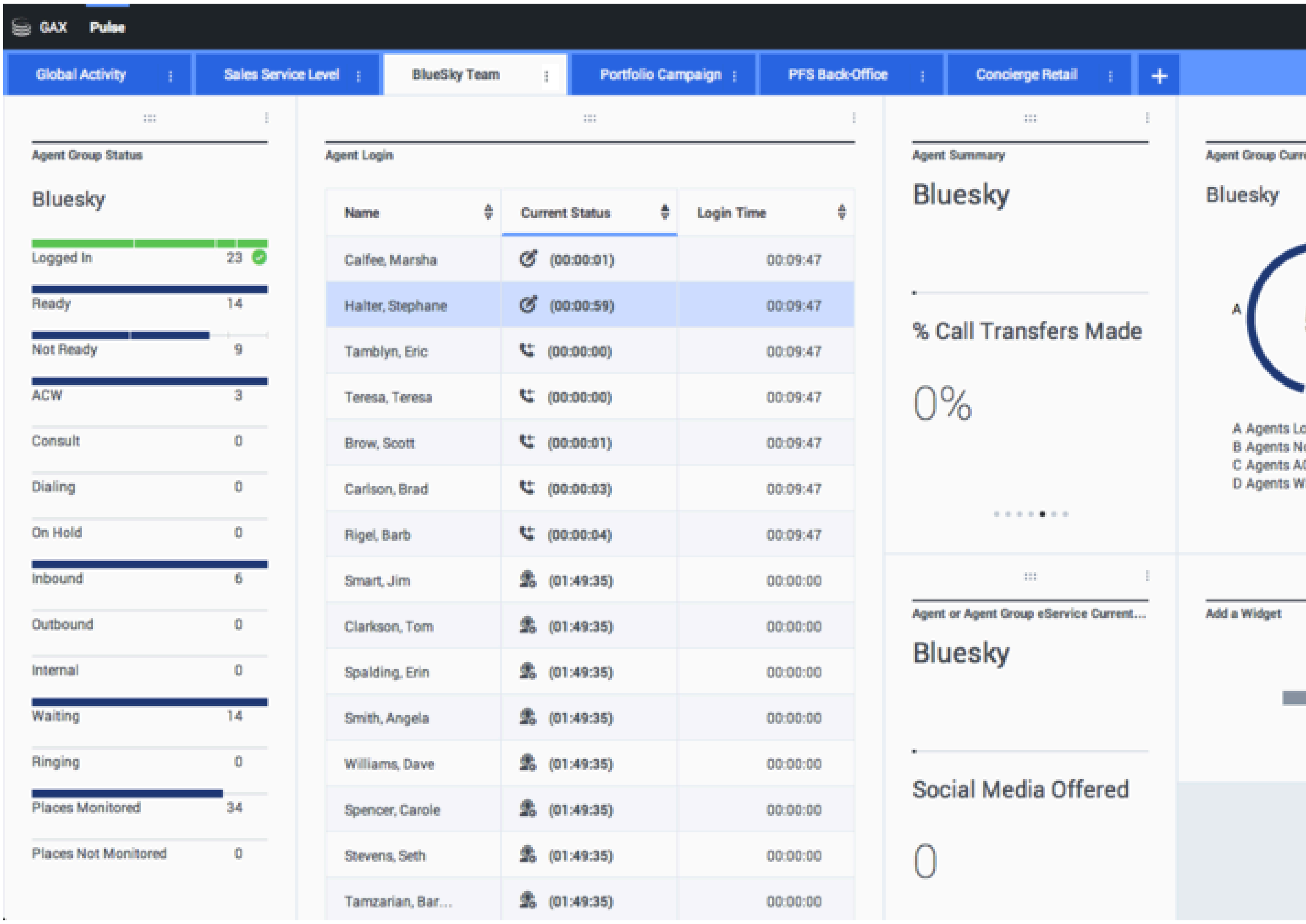
| | |
|---|---|
| A Rayn, Ann | 9815 |
| B Clarkson, Jeremy | 9807 |
| C Gates, Mike | 9781 |
| D Others | 222535 |

Add a Widget

Voice Agent Login with status icon

| Name | Current Status |
|---|---|
| 911 | (617:19:54) |
| Bloch, Joshua | (617:19:54) |
| Brown, Jack | (617:19:54) |

User-defined Sales Team Lead dashboard



Agent Group Status

Bluesky

| | |
|---|---|
| Logged In | 23 ✓ |
| Ready | 14 |
| Not Ready | 9 |
| ACW | 3 |
| Consult | 0 |
| Dialing | 0 |
| On Hold | 0 |
| Inbound | 6 |
| Outbound | 0 |
| Internal | 0 |
| Waiting | 14 |
| Ringing | 0 |
| Places Monitored | 34 |
| Places Not Monitored | 0 |

Agent Login

| Name | Current Status | Login Time |
|---|---|---|
| Calfee, Marsha | (00:00:01) | 00:09:47 |
| Halter, Stephane | (00:00:59) | 00:09:47 |
| Tamblyn, Eric | ☎ (00:00:00) | 00:09:47 |
| Teresa, Teresa | ☎ (00:00:00) | 00:09:47 |
| Brow, Scott | ☎ (00:00:01) | 00:09:47 |
| Carlson, Brad | ☎ (00:00:03) | 00:09:47 |
| Rigel, Barb | ☎ (00:00:04) | 00:09:47 |
| Smart, Jim | (01:49:35) | 00:00:00 |
| Clarkson, Tom | (01:49:35) | 00:00:00 |
| Spalding, Erin | (01:49:35) | 00:00:00 |
| Smith, Angela | (01:49:35) | 00:00:00 |
| Williams, Dave | (01:49:35) | 00:00:00 |
| Spencer, Carole | (01:49:35) | 00:00:00 |
| Stevens, Seth | (01:49:35) | 00:00:00 |
| Tamzarian, Bar... | (01:49:35) | 00:00:00 |

Agent Summary

Bluesky

% Call Transfers Made

0%

Agent Group Curr

Bluesky

A

A Agents Lo
B Agents N
C Agents A
D Agents W

Agent or Agent Group eService Current...

Bluesky

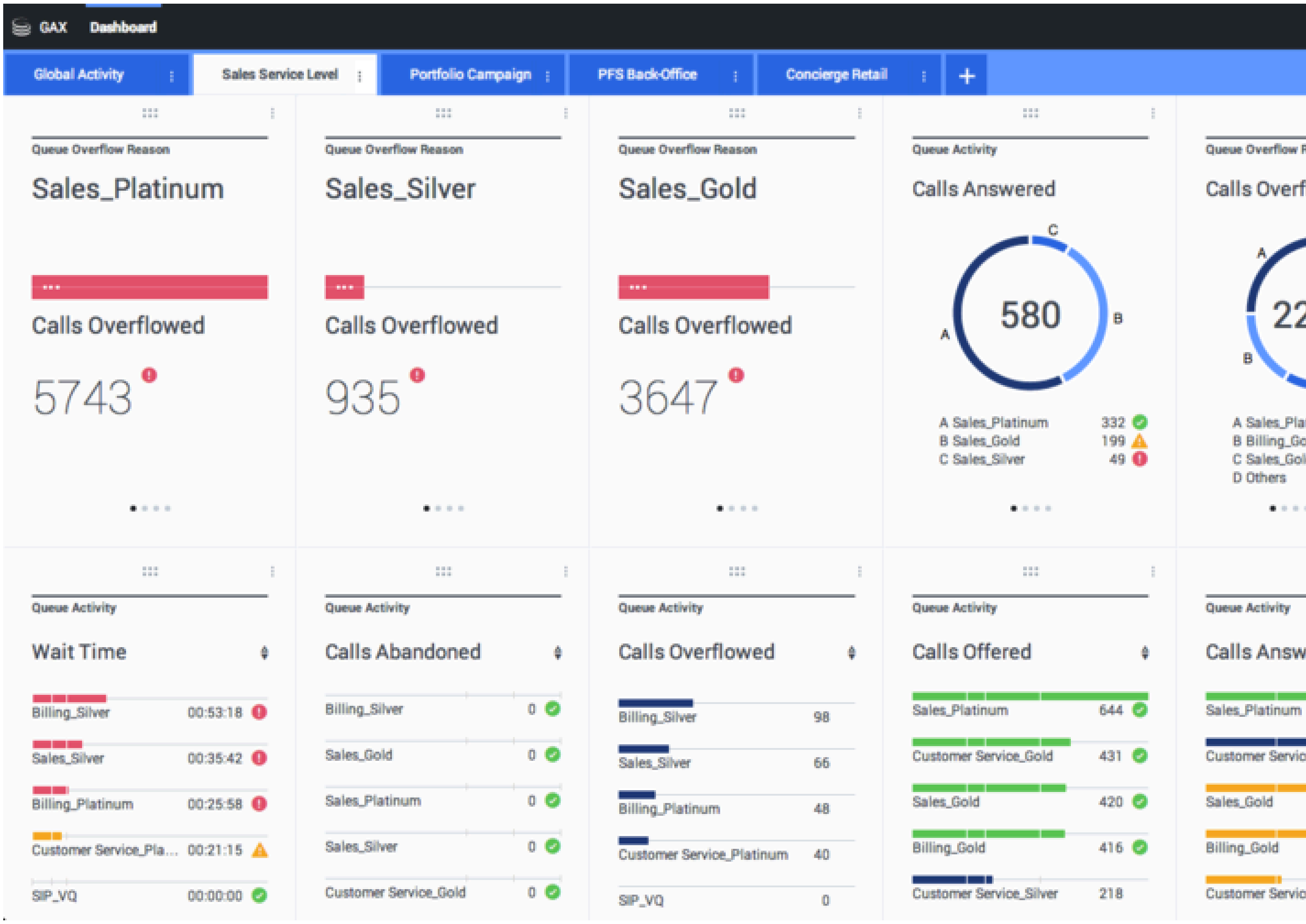
Social Media Offered

0

Add a Widget

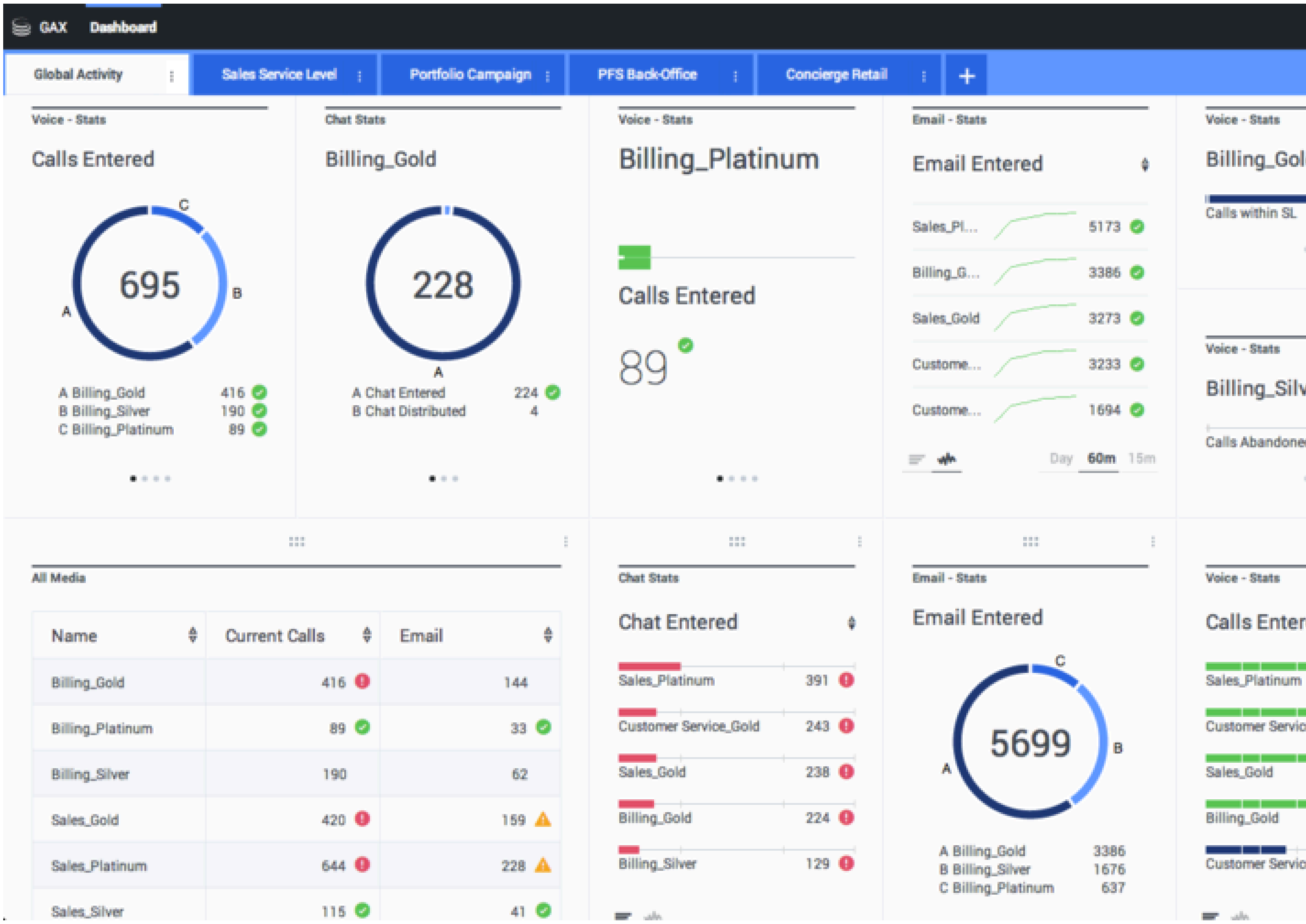
User-defined Sales Team Lead dashboard

2. Example of a Sales Service Level dashboard for Supervisor



User-defined Sales Service Level dashboard for Supervisor

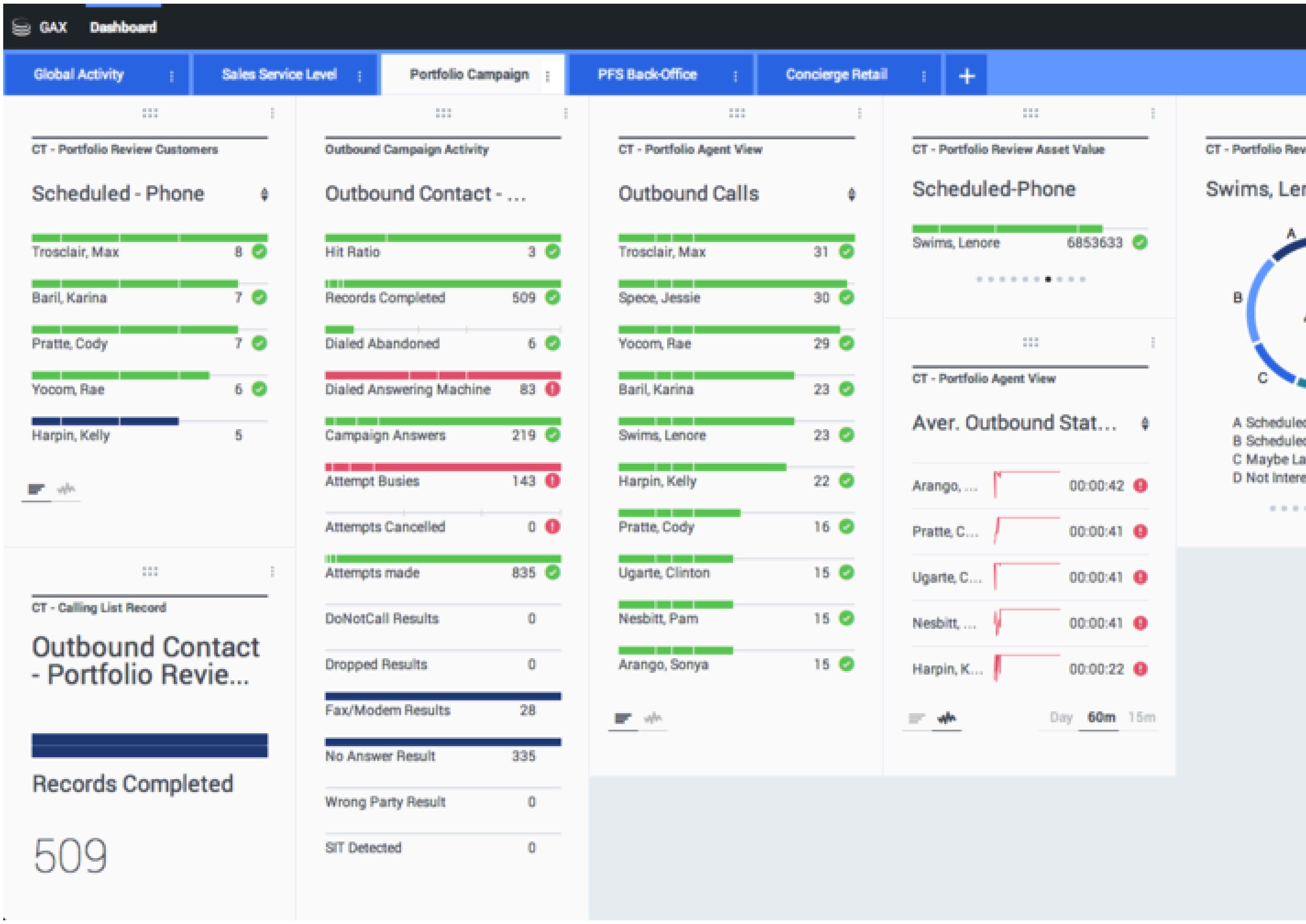
3. Example of a multi-channel dashboard for Supervisor



| Name | Current Calls | Email |
|---|---|---|
| Billing_Gold | 416 | 144 |
| Billing_Platinum | 89 | 33 |
| Billing_Silver | 190 | 62 |
| Sales_Gold | 420 | 159 |
| Sales_Platinum | 644 | 228 |
| Sales_Silver | 115 | 41 |

User-defined multi-channel dashboard for Supervisor

4. Example of an outbound campaign dashboard for Supervisor



CT - Portfolio Review Customers

Scheduled - Phone

| | | |
|---|---|---|
| Trosclair, Max | 8 | ✓ |
| Baril, Karina | 7 | ✓ |
| Pratte, Cody | 7 | ✓ |
| Yocom, Rae | 6 | ✓ |
| Harpin, Kelly | 5 | |

CT - Calling List Record

Outbound Contact - Portfolio Revie...

Records Completed

509

Outbound Campaign Activity

Outbound Contact - ...

| | | |
|---|---|---|
| Hit Ratio | 3 | ✓ |
| Records Completed | 509 | ✓ |
| Dialed Abandoned | 6 | ✓ |
| Dialed Answering Machine | 83 | ! |
| Campaign Answers | 219 | ✓ |
| Attempt Busies | 143 | ! |
| Attempts Cancelled | 0 | ! |
| Attempts made | 835 | ✓ |
| DoNotCall Results | 0 | |
| Dropped Results | 0 | |
| Fax/Modem Results | 28 | |
| No Answer Result | 335 | |
| Wrong Party Result | 0 | |
| SIT Detected | 0 | |

CT - Portfolio Agent View

Outbound Calls

| | | |
|---|---|---|
| Trosclair, Max | 31 | ✓ |
| Spece, Jessie | 30 | ✓ |
| Yocom, Rae | 29 | ✓ |
| Baril, Karina | 23 | ✓ |
| Swims, Lenore | 23 | ✓ |
| Harpin, Kelly | 22 | ✓ |
| Pratte, Cody | 16 | ✓ |
| Ugarte, Clinton | 15 | ✓ |
| Nesbitt, Pam | 15 | ✓ |
| Arango, Sonya | 15 | ✓ |

CT - Portfolio Review Asset Value

Scheduled-Phone

| | | |
|---|---|---|
| Swims, Lenore | 6853633 | ✓ |

CT - Portfolio Agent View

Aver. Outbound Stat...

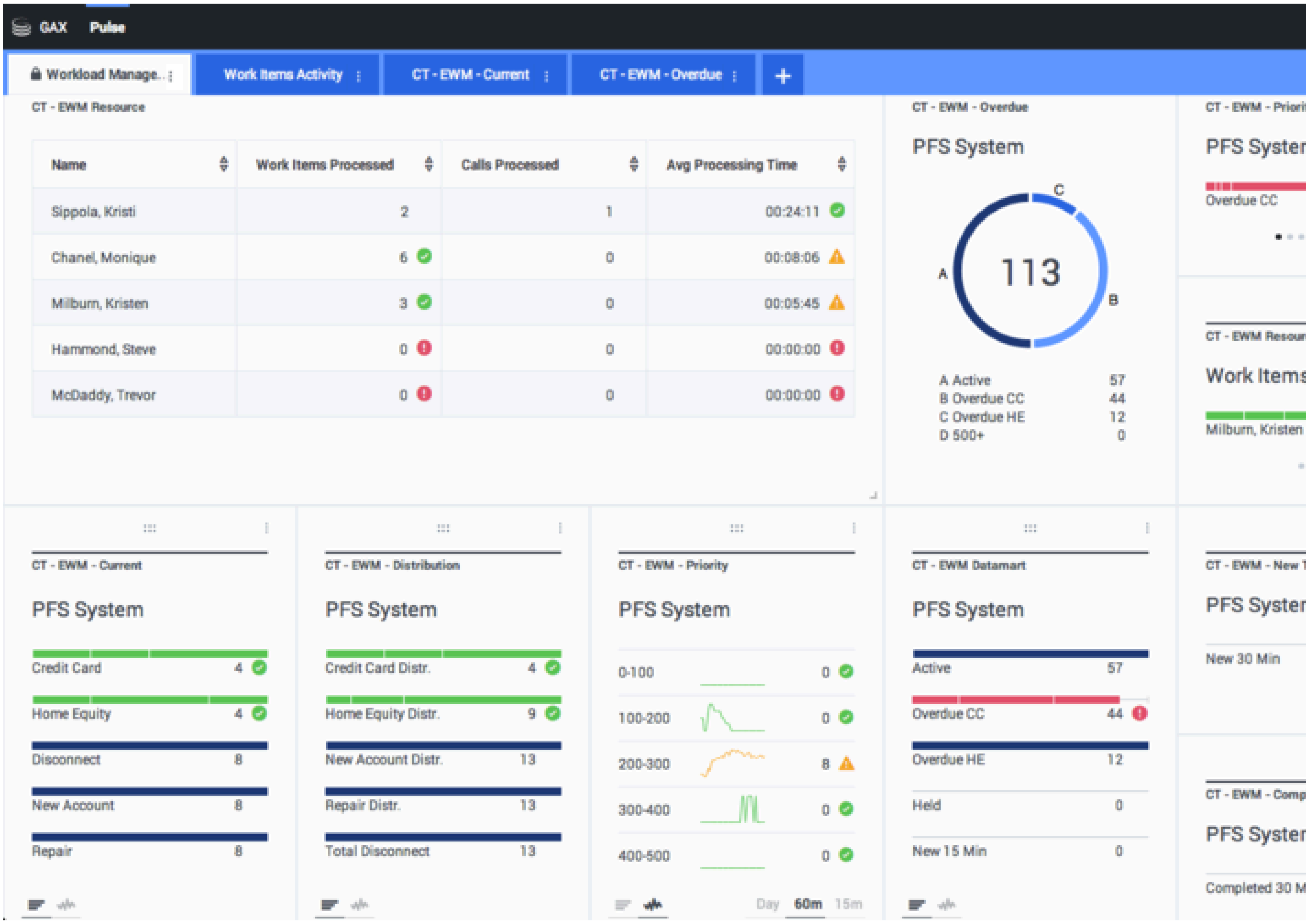
| | | |
|---|---|---|
| Arango, ... | 00:00:42 | ! |
| Pratte, C... | 00:00:41 | ! |
| Ugarte, C... | 00:00:41 | ! |
| Nesbitt, ... | 00:00:41 | ! |
| Harpin, K... | 00:00:22 | ! |

CT - Portfolio Rev

Swims, Ler

User-defined outbound campaign dashboard for Supervisor

5. Example of a back-office dashboard for Supervisor



CT - EWM Resource

| Name | Work Items Processed | Calls Processed | Avg Processing Time |
|---|---|---|---|
| Sippola, Kristi | 2 | 1 | 00:24:11 ✓ |
| Chanel, Monique | 6 ✓ | 0 | 00:08:06 ⚠ |
| Milburn, Kristen | 3 ✓ | 0 | 00:05:45 ⚠ |
| Hammond, Steve | 0 ❗ | 0 | 00:00:00 ❗ |
| McDaddy, Trevor | 0 ❗ | 0 | 00:00:00 ❗ |

CT - EWM - Overdue

PFS System

| | |
|---|---|
| A Active | 57 |
| B Overdue CC | 44 |
| C Overdue HE | 12 |
| D 500+ | 0 |

CT - EWM - Priori

PFS Syster

Overdue CC

CT - EWM Resour

Work Items

Milburn, Kristen

CT - EWM - Current

PFS System

| | |
|---|---|
| Credit Card | 4 ✓ |
| Home Equity | 4 ✓ |
| Disconnect | 8 |
| New Account | 8 |
| Repair | 8 |

CT - EWM - Distribution

PFS System

| | |
|---|---|
| Credit Card Distr. | 4 ✓ |
| Home Equity Distr. | 9 ✓ |
| New Account Distr. | 13 |
| Repair Distr. | 13 |
| Total Disconnect | 13 |

CT - EWM - Priority

PFS System

| | |
|---|---|
| 0-100 | 0 ✓ |
| 100-200 | 0 ✓ |
| 200-300 | 8 ⚠ |
| 300-400 | 0 ✓ |
| 400-500 | 0 ✓ |

CT - EWM Datamart

PFS System

| | |
|---|---|
| Active | 57 |
| Overdue CC | 44 ❗ |
| Overdue HE | 12 |
| Held | 0 |
| New 15 Min | 0 |

CT - EWM - New 1

PFS Syster

New 30 Min

CT - EWM - Comp

PFS Syster

Completed 30 M

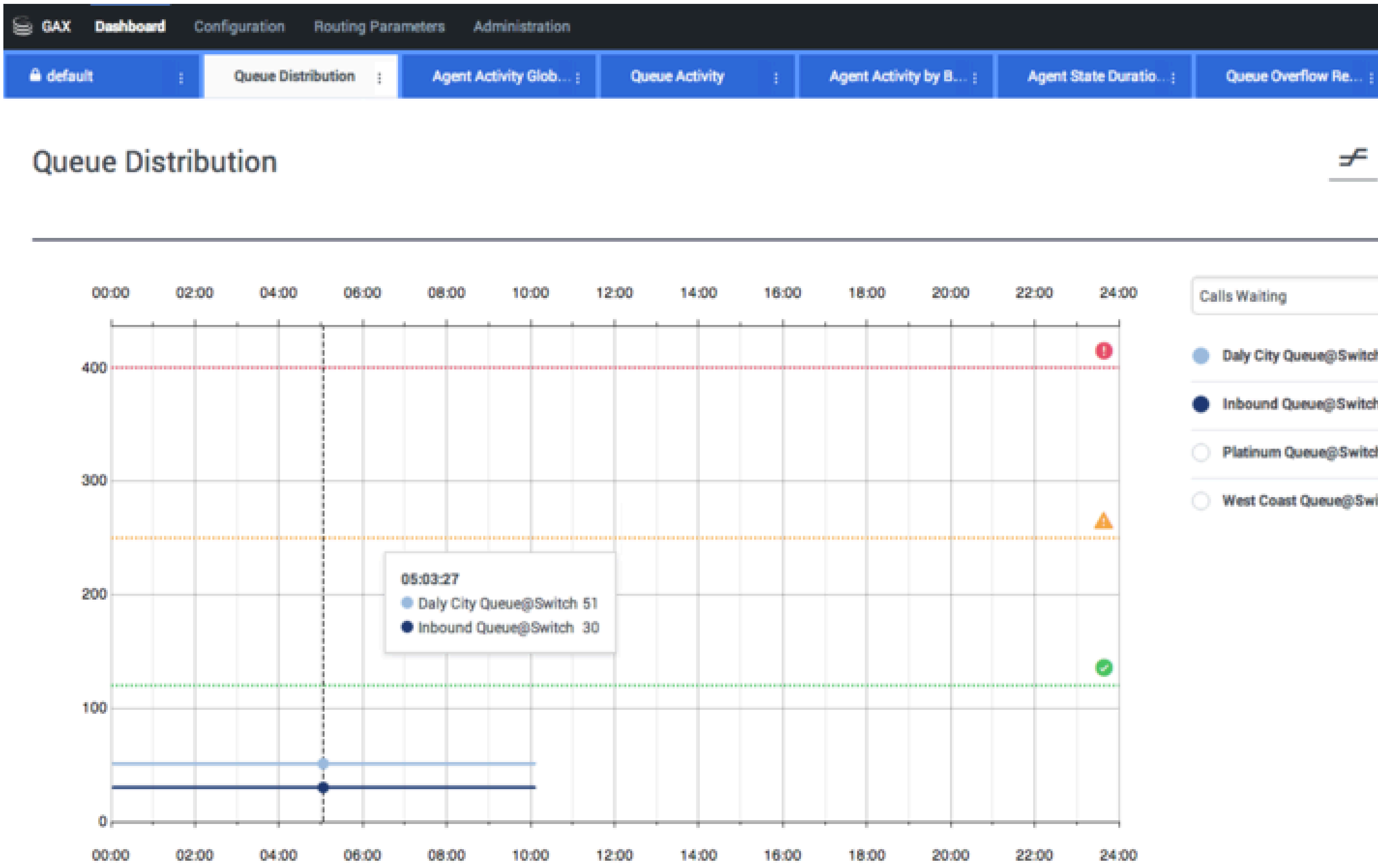
User-defined back-office dashboard for Supervisor

# Expanded Charts

# Expanded Charts

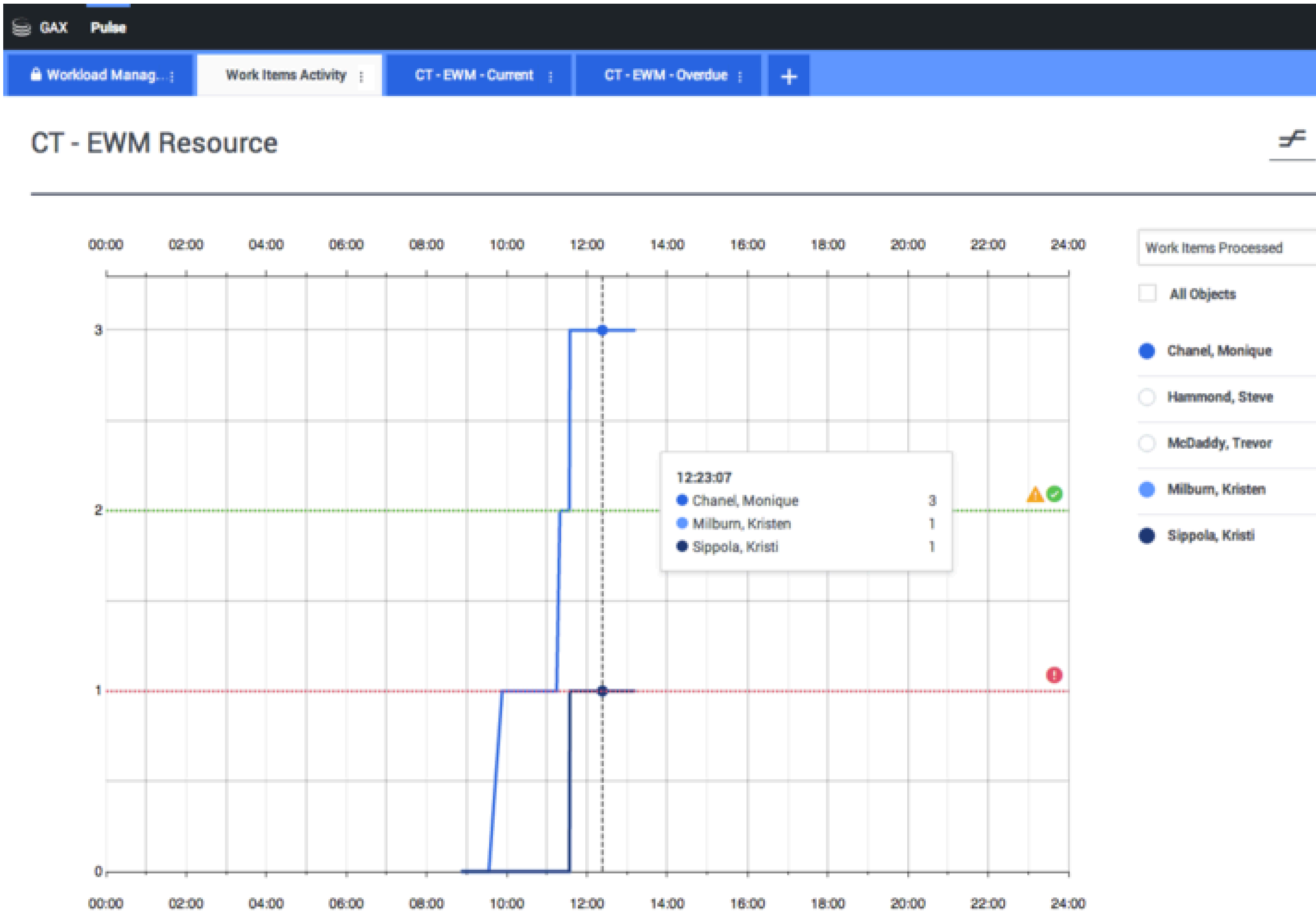
You can expand a widget to view detailed charts: Line; Grouped Bar; Stacked Bar; and Grid. Within the expanded charts, you can use sort options, define objects, and define statistics.

1. Line Chart



05:03:27
Daly City Queue@Switch 51
Inbound Queue@Switch 30

Line Chart

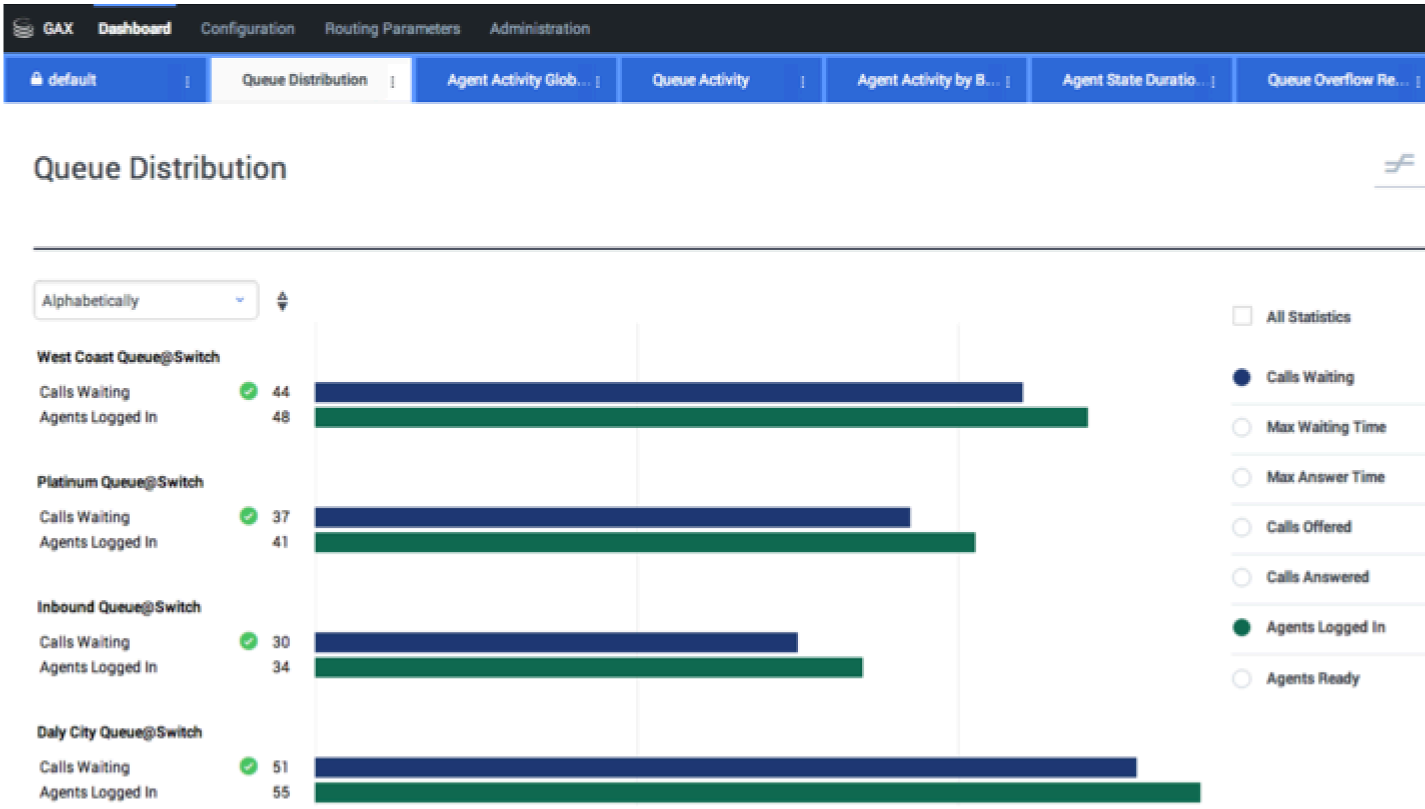


12:23:07

| | |
|---|---|
| ● Chanel, Monique | 3 |
| ● Milburn, Kristen | 1 |
| ● Sippola, Kristi | 1 |

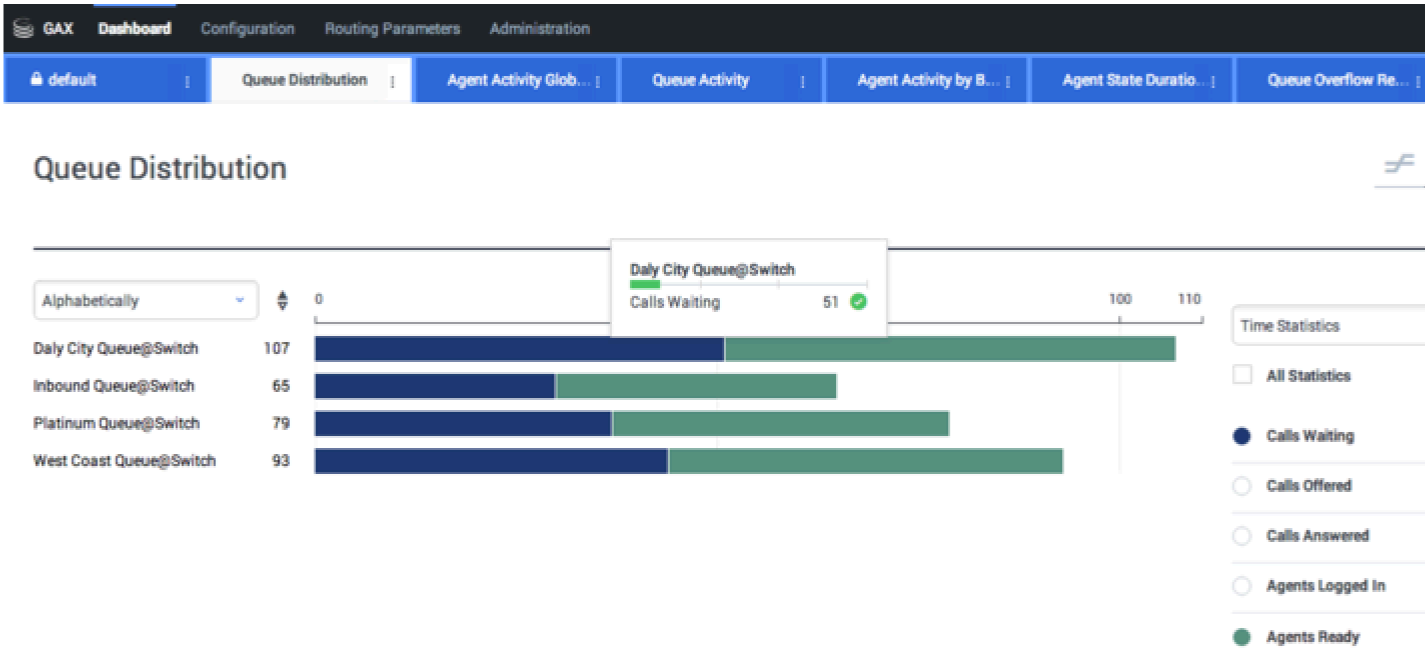
Line Chart

2. Grouped Bar Chart



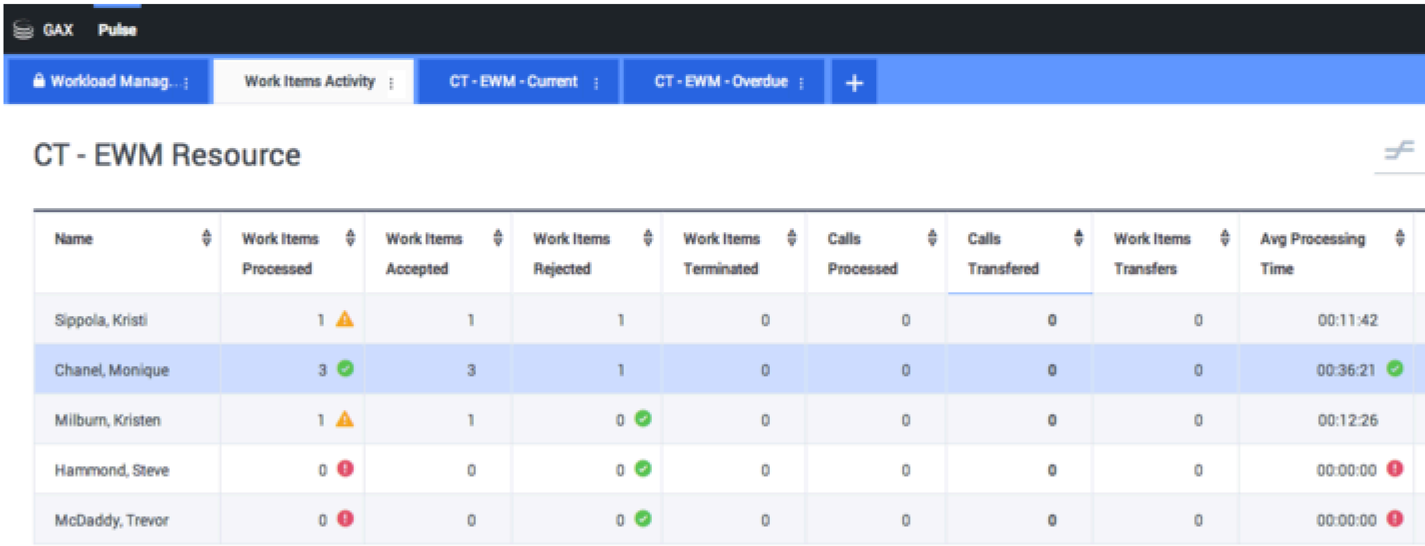
Grouped Bar Chart

3. Stacked Bar Chart



Stacked Bar Chart

4. Grid



CT - EWM Resource

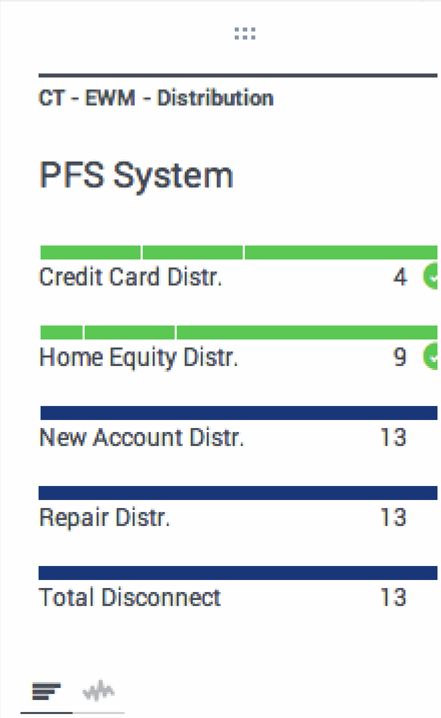
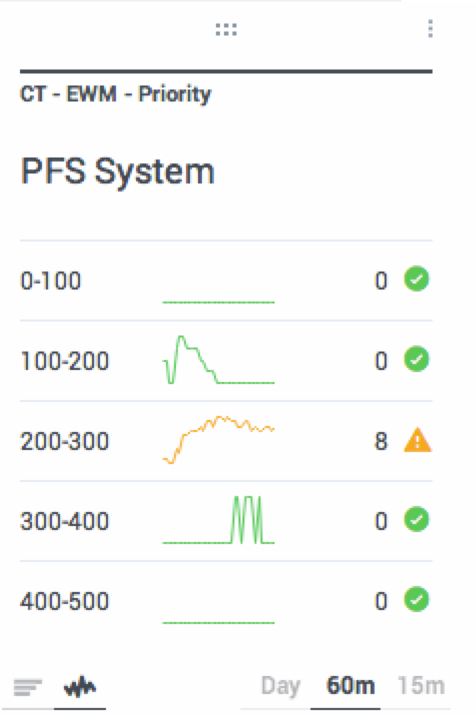
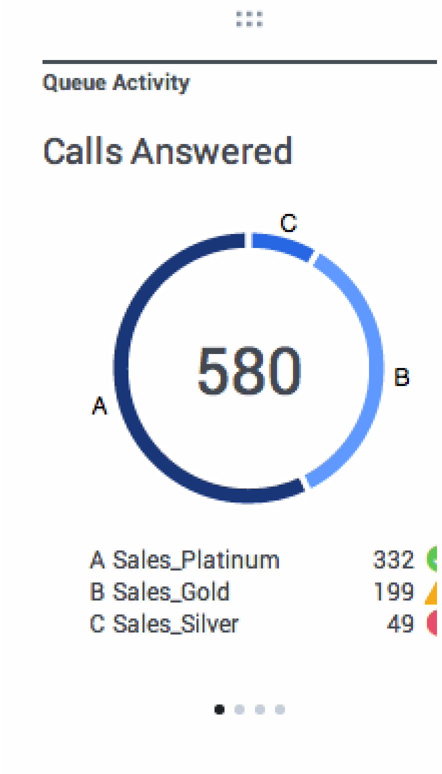
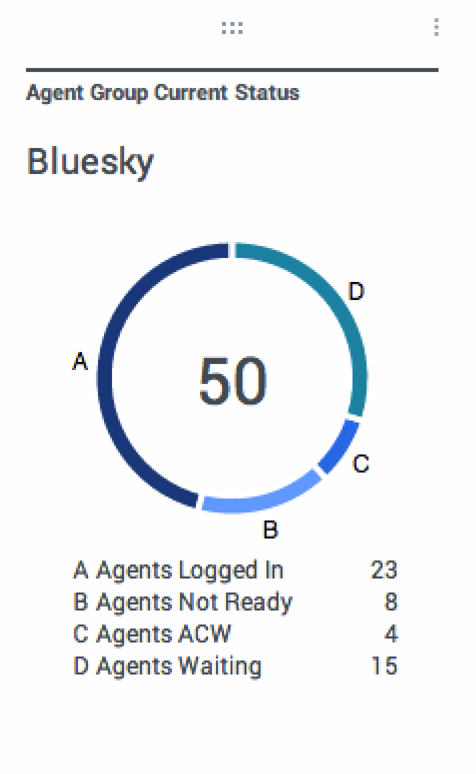
| Name | Work Items Processed | Work Items Accepted | Work Items Rejected | Work Items Terminated | Calls Processed | Calls Transfered | Work Items Transfers | Avg Processing Time |
|---|---|---|---|---|---|---|---|---|
| Sippola, Kristi | 1 ⚠ | 1 | 1 | 0 | 0 | 0 | 0 | 00:11:42 |
| Chanel, Monique | 3 ✓ | 3 | 1 | 0 | 0 | 0 | 0 | 00:36:21 ✓ |
| Milburn, Kristen | 1 ⚠ | 1 | 0 ✓ | 0 | 0 | 0 | 0 | 00:12:26 |
| Hammond, Steve | 0 ❗ | 0 | 0 ✓ | 0 | 0 | 0 | 0 | 00:00:00 ❗ |
| McDaddy, Trevor | 0 ❗ | 0 | 0 ✓ | 0 | 0 | 0 | 0 | 00:00:00 ❗ |

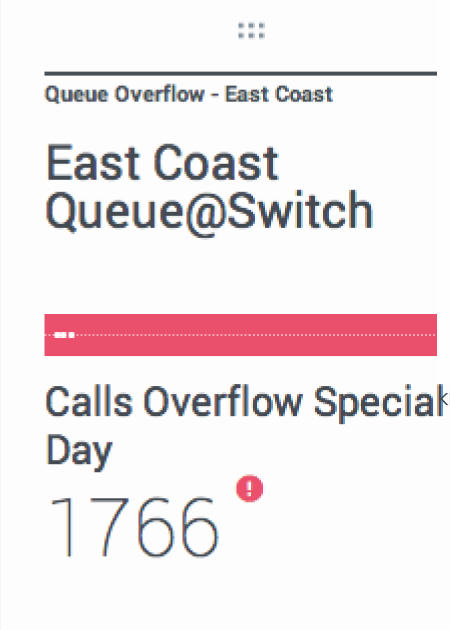
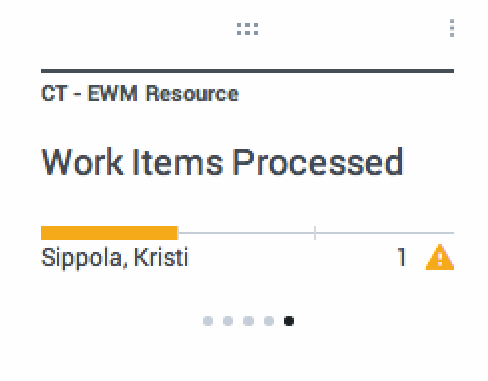
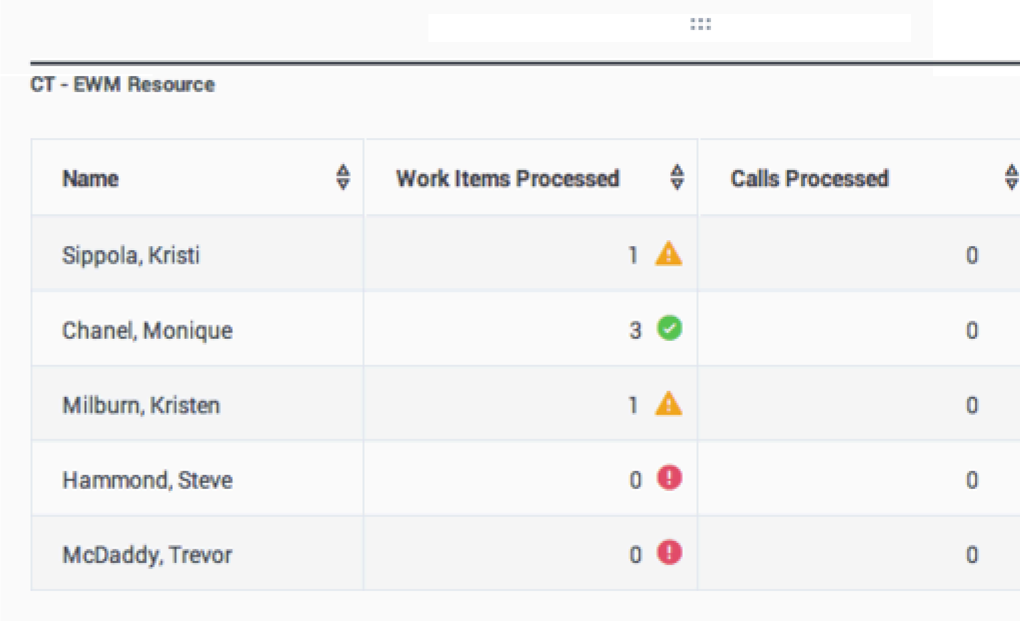
Grid

## Widgets

## Widgets

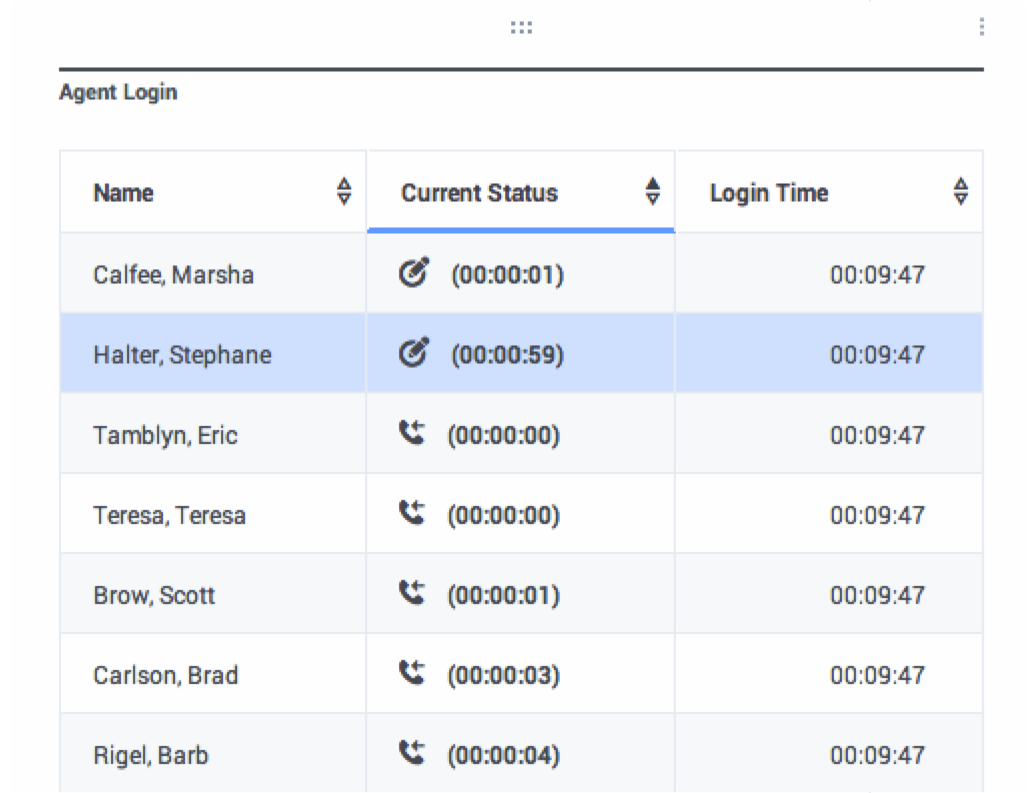
On the Pulse dashboard, statistics and objects can be displayed within different type of user-defined widgets: Donut; Grid; Key Performance Indicator (KPI); and List. You can view and select additional details and options by expanding a widget to a dashboard tab.

| Widget | Examples | |
|---|---|---|
| **List Widget**<br>This widget displays either one statistic for many objects or many statistics for one object. Depending on the reference selected, the Headline type option might be available for this widget type. | <br>CT - EWM - Distribution<br>PFS System<br>Credit Card Distr. 4 ✓<br>Home Equity Distr. 9 ✓<br>New Account Distr. 13<br>Repair Distr. 13<br>Total Disconnect 13<br>List Widget | <br>CT - EWM - Priority<br>PFS System<br>0-100 0 ✓<br>100-200 0 ✓<br>200-300 8 ⚠<br>300-400 0 ✓<br>400-500 0 ✓<br>Day 60m 15m<br>List Widget |
| **Donut Widget**<br>This widget displays either:<br>• one statistic for four specific objects<br>• one statistic for three top objects and a sum the remaining objects<br>• one object with the values of four defined statistics<br>Depending on the reference selected in the Cycle By option, a carousel can be defined to display additional several items. | <br>Queue Activity<br>Calls Answered<br>580<br>A Sales_Platinum 332<br>B Sales_Gold 199<br>C Sales_Silver 49 | <br>Agent Group Current Status<br>Bluesky<br>50<br>A Agents Logged In 23<br>B Agents Not Ready 8<br>C Agents ACW 4<br>D Agents Waiting 15 |

| Widget | Examples | |
|---|---|---|
| | Donut Widget | Donut Widget |
| **KPI widget**<br>This widget displays either one statistic for several objects or several statistics for one object, depending on the value of the Cycle By option. The Cycle By option is available if the widget has objects selected individually, not by group. | <br>Queue Overflow - East Coast<br>East Coast Queue@Switch<br>Calls Overflow Special Day<br>1766<br>KPI Widget | <br>CT - EWM Resource<br>Work Items Processed<br>Sippola, Kristi 1<br>KPI Widget |
| **Grid widget**<br>This widget displays a grid listing objects in rows with statistics in columns. | <br>Grid Widget | |

CT - EWM Resource

| Name | Work Items Processed | Calls Processed |
|---|---|---|
| Sippola, Kristi | 1 | 0 |
| Chanel, Monique | 3 | 0 |
| Milburn, Kristen | 1 | 0 |
| Hammond, Steve | 0 | 0 |
| McDaddy, Trevor | 0 | 0 |

| Widget | Examples |
|---|---|
| |  Grid Widget |

Agent Login

| Name | Current Status | Login Time |
|---|---|---|
| Calfee, Marsha | ✓ (00:00:01) | 00:09:47 |
| Halter, Stephane | ✓ (00:00:59) | 00:09:47 |
| Tamblyn, Eric | ☎ (00:00:00) | 00:09:47 |
| Teresa, Teresa | ☎ (00:00:00) | 00:09:47 |
| Brow, Scott | ☎ (00:00:01) | 00:09:47 |
| Carlson, Brad | ☎ (00:00:03) | 00:09:47 |
| Rigel, Barb | ☎ (00:00:04) | 00:09:47 |

# Genesys-provided Templates

## Genesys-provided Templates

The following is a list of Genesys-provided templates:

| Template | Description |
|---|---|
| Agent Group Status | Presents the current number of agents in the various states. |
| Agent KPIs | Presents reports with KPIs of agent group in a contact center. |
| Agent Login | Presents an agent's properties and login information. |
| Campaign Activity | Monitors the activity associated with outbound campaigns. |
| Campaign Callback Status | A report presenting information related to |

| Template | Description |
|---|---|
| | campaign initiated callbacks. |
| Campaign Group Activity | Monitor the activity associated with outbound Campaign Groups. |
| Campaign Group Status | Monitor the current state and durations associated with outbound campaign group activity. |
| Email Agent Activity | A report presenting agent or agent group activity as it relates to the processing of Email type contacts. |
| Email Queue Activity | A queue report presenting an overview of current or near real-time activity in the individual email queues. |
| eServices Agent Activity | Monitors agent group KPIs related to eServices (chat, email, SM) media to monitor and see if behavior problems exist that need to be addressed. |
| eServices Queue KPIs | A queue report presenting an overview of current or near real-time activity for eServices channels. |
| IWD Agent Activity | A report presenting agent or agent group activity as it relates to the processing iWD work items type contacts. |
| IWD Queue Activity | A queue report presenting an overview of current or near real-time activity associated with the iWD queues. |
| Queue KPIs | Presents call activity statistics associated with the queues. |
| Queue Overflow Reason | Presents reasons why calls were cleared from queues. |

# Pulse Architecture and Components

Pulse is a Genesys Administrator Extension (GAX) plug-in application that enables at-a-glance views of real-time contact center statistics within the GAX graphical user interface. Pulse uses widgets to display user-defined List, Donut, Key Performance Indicator (KPI), or Grid charts of statistics for objects.

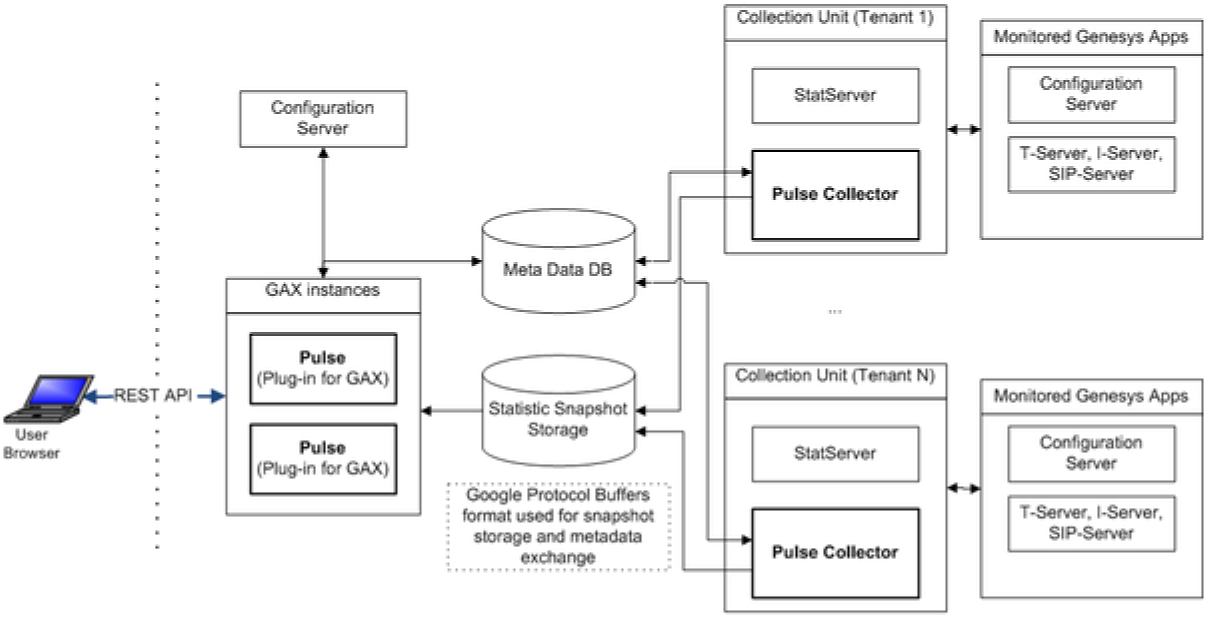
Using Pulse you can:

- Create widgets from predefined and user-defined templates for a fast and easy text or graphical presentation of selected or user-defined object statistics.
- Monitor the current state and activity of contact center objects to help make decisions about staffing, scheduling and call routing strategies.

## Architecture

### Pulse Architecture

Major aspects of Pulse are shown in the following Pulse Architecture diagram:



**Notes:**

- One collection unit can serve multiple tenants, but you can have only one collection unit for each tenant.
- Each collection unit supports Genesys High Availability (for example, primary and backup Stat Servers and Pulse Collectors)

## Pulse Collector

Pulse Collector is a background near-realtime statistical data collection and processing engine. It performs the following activities:

- Reads the metadata from the Pulse database upon startup and whenever changes are made to report definitions in Pulse.
- Uses the report definitions stored in the Pulse database to determine which statistics and objects to include.
- Creates snapshots with current data from Stat Server and formula-based statistics calculated by Pulse Collector, on the specified file system for reference by Pulse.
- Maintains a constant connection with Configuration Server to retrieve changes, additions, and deletions to configuration objects.

## Pulse

Pulse is a GAX plug-in that performs the following activities:

- Handles user authentication and permissions validation.
- Filters and delivers report data according to the permissions and tenancy of the user who is requesting the data.

Displays report content in widgets, such as the listing and content of reports.

- Saves the report definitions to the Pulse Database, which it shares with Pulse Collector.

You can have only one Pulse application for each GAX instance. For configuration, Pulse requires a `[pulse]` section and is added to the options of the GAX Application object. Pulse must have its own database access point, but otherwise uses Management Framework and other connections from GAX.

Pulse uses the existing GAX client architecture, which is described in the Genesys Administrator Extension Deployment Guide.

See the Pulse User's Guide to learn how to operate this user interface. This is also accessible from within the software, when you click help.

# Framework Components

# Genesys Framework Components

Pulse interacts with several products within the Genesys Framework to provide real-time snapshots of contact center data.

## Configuration Server

Configuration Server provides the following data to Pulse Collector:

- Information about the existence of contact center objects (for example, tenants, agents, places, calling lists, or campaigns)
- Statistical parameters (for example, time ranges, time profiles, filters, and statistical types)
- Information about changes to contact center objects (for example, a deleted agent, a renamed queue, or a new Agent Group).

Pulse Collector uses this data to provide content for Pulse.

Both Pulse and Pulse Collector connect to Configuration Server in order to retrieve their own configuration.

## Stat Server

Stat Server tracks information about customer interaction networks consisting of one or more contact centers in conjunction with one or more computer networks. Stat Server receives information from one or more T-Server or SIP Server applications and converts the accumulated data for directory numbers, agents, agent groups, and so on, into statistical information.

As a client of Stat Server, Pulse Collector requests statistics for objects belonging to particular reports. Stat Server supplies both information about the interactions that the objects that handle as well as noninteraction-related data about the objects themselves (for example, agent status). Pulse Collector returns information for all `stattypes` that are configured in the options of all Stat Servers to which Collector is connected.

Refer to the Stat Server User’s Guide and Stat Server Deployment Guide for information about Stat Server.

## DB Server

DB Server is the Genesys component that handles database requests from multiple client processes. DB Server provides a single interface from the clients to a variety of database engines, including Microsoft SQL Server, Oracle and PostgreSQL. As a client of DB Server, Pulse Collector reads information about active widgets and updates the layout statuses, when layout status changes occur within the Pulse Collector.

Refer to the [Framework 8.1 DB Server User’s Guide] for information about DB Server.

# Deploying Pulse Collector and Pulse

> **Important**
>
> - This Deployment Guide provides instructions for a new installation of Pulse. To migrate from an earlier version of Pulse, start with the Deployment Procedure.
> - In these procedures, use the instructions provided in Genesys Administrator Extension (GAX) Help or the Framework 8.x Deployment Guide, and add the object-specific configuration requirements listed here.

## 1. Confirm Software Requirements.

## Confirm Software Requirements

You deploy Pulse as a GAX plug-in on a web application server to be accessed using a web browser through the GAX user interface. No additional Genesys software is needed on the client machine.

The following are prerequisites for Pulse deployment:

- GAX release 8.5.000.44 or higher must be installed and configured. See the GAX Deployment Guide for details.
- Tomcat is no longer a prerequisite to use GAX. For those who choose to use Tomcat instead of Jetty, refer to the GAX Deployment Guide for additional information.
- Genesys DB Server must be release 8.1.300.05 or higher. Refer to the Framework 8.0 DB Server User's Guide for details.
- Genesys Deployment Agent (GDA) must be installed on the computer on which you install Pulse and Pulse Collector if you plan to install Pulse by using GAX. GDA is required to deploy solution definitions and IPs through GAX. See the GAX Deployment Guide for details.
- Your Relational Database Management System (RDBMS) must be up and running. The 8.5 release of Pulse supports the following relational database platforms:
  - Microsoft SQL Server 2008, 2012
  - Oracle 11g, 12c
  - PostgreSQL 9.0
- The computer on which you install Pulse must be running one of the following:
  - Windows Server 2008 R2 (64-bit), 2012 (64-bit)
  - Red Hat Enterprise Linux AS 6.x (64-bit), with Updates from RHN enabled

- If you use Red Hat Enterprise Linux AS, you must also install the following libraries to ensure backward compatibility for applications: compat-db
    - compat-expat1
    - compat-glibc
    - compat-libf2c-34
    - compat-libgcc-296
    - compat-libgfortran-41
    - compat-libstdc++-295
    - compat-libstdc++-296
    - compat-libstdc++-33
    - compat-libtermcap
    - compat-openldap
    - openssl098e

  For more information, see linuxtopia.org.
- If you use Windows, you must also install Microsoft Visual C++ 2010 Redistributable Package (x86 for Collector x86 and x64 for Collector x64).
- Although not required for deployment, you must have Stat Server release 8.1.2 or higher installed for basic operation.

# 2. Prepare the Pulse Database.

## Prepare the Pulse Database

In High Availability (HA) configurations, configure both Pulse applications to use the same database.

### Microsoft SQL Server

1. Create a new empty Microsoft SQL Server database.
2. Create a new Microsoft SQL Server user account.
3. Set the new database as default database for the user.
4. Grant the new user sufficient privileges for the new database. User must be able to create database objects and select, insert, update, and delete data in tables.
5. Run the statement below for the Pulse database (DB). Replace `<Pulse DB>` with the name of the Pulse DB and make sure that there are no other connections to the Pulse DB when you run this statement: `ALTER DATABASE <Pulse DB> SET READ_COMMITTED_SNAPSHOT ON;`

### Oracle

1. Create a new user/schema to be used by Pulse. The user must have `RESOURCE` and `CREATE VIEW` privileges.

### PostgreSQL

1. Create a new empty PostgreSQL database.
2. Create a new PostgreSQL user account.
3. Set the new database as default database for the user.
4. Grant the new user sufficient privileges for the new database. User must be able to create database objects and select, insert, update, and delete data in tables.

# 3. Deploy Pulse Collector.

## Deploying Pulse Collector

To deploy Pulse Collector, which connects directly to Stat Server to collect statistics for contact center objects. Pulse Collector also connects to DB Server, through which it accesses the database where reporting layouts are stored.

1. Upload Pulse Collector Installation Package and Template:
   a. In GAX, go to **Administration > Installation Packages** and click on the plus sign.
   b. Select **Installation Package Upload (template uploaded separately)** and click **Next**.
   c. For **Upload a Package**, select the zipped file that contains the Pulse Collector Installation Package (for example, `\PulseCol\version\linux\b1\ip`, which has the Installation Package description file).
   d. For **Upload an XML template**, select the XML Template file (Collector.xml from the Templates Installation Package directory).
   e. For **Upload an APD template**, select the APD Template file (Collector.apd from Templates Installation Package directory).
   f. Click **Finish**.
2. Deploy the Pulse Collector Installation Package and Template:
   a. Click on the Pulse Collector Installation Package to open the Properties tab.

> Important

> The Pulse Collector Installation Package status should be Complete.

b. Click on the related icon and choose **Install** to open the IP Deployment Wizard.

c. Fill required fields and finish the installation.

> Important
>
> The InstallPath should point to an empty folder.

> Tip
>
> You can also install Pulse Collector directly on the server by executing the Pulse Collector installation procedure from the Pulse Collector installation package, then uploading the Pulse Collector template (Collector.apd) using Configuration Manager.

3. Optional: To change the default options for Pulse Collector in GAX, open the Pulse Collector `Application` object modify the values of configuration options described in Pulse Collector Application Object.
4. For a high-availability (HA) deployment, repeat step 2 to install the backup instance to separate folder. You need a Pulse Collector for each instance of the Pulse application.
5. In GAX, add the Pulse Collector `Application` object to the connections of your GAX `Application` object.
6. For a load-balanced environment configuration with two GAX applications and Pulse plug-ins, associate one GAX with the primary Pulse Collector and the second GAX with the backup Pulse Collector as follows:

   a. Add the primary Pulse Collector `Application` object to the connections of the first (or primary) GAX `Application` object.

7. Add the backup Pulse Collector `Application` object to the connections of the second (or backup) GAX `Application` object.
8. Add the backup Pulse Collector `Application` object to the **General** tab of the primary Pulse Collector `Application` object as a backup server.
   **Note:** One GAX pair can be connected to multiple Pulse Collector HA pairs. In this case, you must associate all primary Pulse Collectors with one GAX, and all the backup Pulse Collectors with the other GAX.

- Create a new Database Access Point (DAP), which is necessary for connectivity to the Pulse database through the DB Server:

  a. Select the Default connection type.

b. Specify Database Server `Application` object in **DB Server** field.

c. Specify the connectivity parameters for your RDBMS.

d. On the **Ports** tab, change the default port value to the value of your DB Server port.

- In the connections of the Pulse Collector `Application`, add the DAP that is to be used by Pulse Collector.
- In the connections of the Pulse Collector `Application`, add the primary Stat Server application that is to be used by Pulse Collector.
- Add the Tenant objects to the Tenants tab for all Pulse Collectors that you plan to monitor in Pulse.

## 4. Deploy Pulse.

## Deploying Pulse

Configure the necessary objects required by Pulse using GAX.

1. Upload Pulse Installation Package and Template:
   a. In GAX, go to **Administration > Installation Packages** and click on the plus sign.
   b. Select **Installation Package Upload (template uploaded separately)** and click **Next**.
   c. For **Upload a Package**, select the zipped file that contains the Pulse Installation Package (for example, `\Pulse\version\linux\b1\ip`, which has the Installation Package description file).
   d. For **Upload an XML template**, select the XML Template file (Pulse.xml from the Templates Installation Package directory).
   e. For **Upload an APD template**, select the APD Template file (Pulse.apd from Templates Installation Package directory).
   f. Click **Finish**.
2. Deploy the Pulse Installation Package and Template:
   a. Click on the Pulse Installation Package to open the **Properties** tab.

> Important
>
> The Pulse Installation Package status should be Complete.

   b. Click on the related icon and choose **Install** to open the IP Deployment Wizard.
   c. Fill required fields and finish the installation.

### Important

- For IPCommon, the InstallPath should be an empty folder for the Pulse installation (for example, C:\123).
- When installing Pulse on Linux, a second path leading to the GAX installation (the field is called GAX directory) must be specified in addition to InstallPath.

### Tip

You can also install Pulse directly on the server by executing the Pulse installation procedure from the Pulse installation package.

If the installation procedure fails to find the GAX installation, you can manually copy the following files to enable Pulse in GAX: `pulse.jar`, `pulse-proto.jar`, and `protobuf-java-2.5.0.jar` from the Pulse installation: `<GAX root>/webapp/WEB-INF/lib/` on Linux or `<GAX root>\webapp\WEB-INF\lib` on Windows.

3. Optional: Configure the [Pulse] options on **Application Option**s tab for the GAX `Application` object:

### Important

During startup, GAX looks for all options that are required for Pulse operation and adds them to the GAX `Application` object if they are not explicitly configured. If a required option is either not configured or specifies an invalid option value, GAX uses the option’s default value.

4. Create a new Database Access Point, which is necessary for connectivity to the Pulse database:
   a. On the **General** tab, set the **Connection Type** to **JDBC** and specify the following field values:
      - Role = Main
      - Debug = false
      - JDBC Query Timeout: 15
      - Case Conversion: any

   b. On the **Ports** tab, change the value of the default port to the value of your RDBMS port.
   c. On the **Application Options** tab, add the role configuration option with its value set to pulse in the

GAX section to identify this Database Access Point as the database schema created for Pulse.

d. For a PostgreSQL database, add the postgre_71_compatible configuration option with its value set to false in the GAX section on the **Application Options** tab.

Refer to the Framework 8.0 DB Server User's Guide for additional information about configuring this application.

- Add the Database Access Point to the connections of your GAX `Application` object.
- Navigate to **Configuration > Environment > Application**, select the GAX Application, **Permissions** tab and configure the SYSTEM account to have Read, Update, and Execute permissions.
- For a High Availability (HA) deployment,

a. Complete Steps 2-6, for each GAX `Application` object to be used with the Pulse plug-in.

b. Each GAX application should have the other GAX application in its connections and there must be no relation between the GAX applications as Backup Servers.

c. Navigate to GAX application object properties and select Application options tab to create a new [pulse] section.

d. Set the access_url option in the [pulse] Section on the **Application Options** tab of the GAX `Application` object for each instance of GAX.

> Important
>
> Both Pulse instances must use the same database.

- From the user account created when you prepared the Pulse database, execute the SQL statements in the appropriate initialization script deployed during installation (scripts folder)—either:
  - pulse_init_mssql.sql
  - pulse_init_oracle.sql
  - pulse_init_postgres.sql
- Restart GAX. See the GAX Deployment Guide for information about how to start GAX.

## 5. Configure the Stat Server Application object.

## Configuring the Stat Server `Application` object

> Important
>
> Stat Server requires Java Environment configuration for several

> templates to function properly. See the Stat Server Deployment Guide for details. Without this additional configuration, statistics in some Queue templates (Email Queue Activity, eServices Queue KPIs, IWD Queue Activity) will not work.

1. Set the `accept-clients-in-backup-mode` option in the [statserver] section on **Application Options** tab to yes for both the primary and backup Stat Server `Application` objects.

   > Important
   >
   > This option is required even if there is no backup application specified for the Stat Server application.

2. Update `stattypes` specified in the pulse_statistics.cfg file within the scripts folder in Pulse installation to use the particular social media that is configured in your eServices solution (`facebook` is used in default file version). See eServices documentation for more details.
3. You must import pulse_statistics.cfg file to both the primary and backup Stat Server `Application` objects to create the stattypes that Pulse should monitor. To monitor the file:
   a. Click **More** on the **Application Options** tab.
   b. Select **Import**, uncheck **Override**, and browse the file.

   > Important
   >
   > - To calculate the **% Ready Time** in the Queue KPIs template, set the `queue-use-pseudo-actions` option in the `[statserver]` section of Stat Server `Application` object to `false`.
   > - Some Stat Server filters used in Pulse templates rely on certain user data or reasons attached to the call (for example, `VoiceCall_No_Wait`, `ReasonLunch`, and others that have `PairExists` in their definition).
   >
   >   You may need to adjust definitions of these filters to use `Attached Data` or Reasons according to your environment or adjust your routing strategies or desktop application to attach data used by those statistics. Otherwise, statistics that rely on these filters will show 0 (zero).

4. Restart both Stat Server applications.

## 6. Configure user access.

## Configuring User Access

1. In GAX, navigate to **Configuration > Accounts > Roles** and create a new Role object to provide access to Pulse functionality.

   > **Important**
   >
   > When creating a new Role, a dialog box with Role Template Selection appears. Do not select a template (leave the selection empty). Press **OK**.

   a. Define the privileges granted by the Role on the **Assigned Privileges** tab in the Pulse section.

   **[+] Privilege Details**

   Privileges manage the following item access:

   - default dashboard
   - templates
   - widget definition, including objects, statistics, and refresh rate
   - widget display options, including widget type, statistics to display, and alerts
   - statistical data provided through the widget
   - custom dashboard

   **Privileges Defined:**

   - **Pulse Read Layout**—See data provided by a widget. Configure the custom dashboard, including reset it to Default.
   - **Pulse Update Layout**—Edit widget display options.
   - **Pulse Write Layout**—Remove, create, clone, and edit widgets.
   - **Pulse Write Template**—Remove, create, clone, and edit custom templates. Change the default dashboard.

   > **Important**
   >
   > Users without permission to edit their own Person object cannot save their Custom dashboard configuration. Users without permission to edit both their own Person object and their Tenant object cannot save the Default dashboard configuration. For example, such a user can make changes, such as creating widgets, but cannot save the changes. The next time the page is loaded, the changes are lost.

Pulse privileges use GAX logic, so the high-level privileges do not include lower-level privileges. For example, to configure role with full control access, you have to assign all four privileges to it.

For role members to create Widgets, you must assign **Pulse Read Layout** and **Pulse Update Layout** in addition to **Pulse Write Layout**.

> Important
>
> You must assign at least the **Pulse Read Layout** privilege to each Role object.

b. Assign the Role to Persons and Access Groups in the Role Members section as required.

2. Update the GAX User to have the following permissions:
   - Read and execute permissions on the client application (usually called `default`)
   - Read and execute permissions on Tenant
   - Read and update permission on their own Person object

> Important
>
> Users without permission to edit their own Person object cannot save their Custom dashboard configuration. Users without permission to edit both their own Person object and their Tenant object cannot save the Default dashboard configuration. For example, such a user can make changes, such as creating widgets, but cannot save the changes. The next time the page is loaded, the changes are lost.

a. Provide the user permission to modify the tenant configuration object.

   i. Select Configuration.

   ii. From the Environment pane, choose Tenants.

   iii. Click to open the required Tenant to view its properties.

   iv. Select Permissions tab.

   v. Add the Person or Access group containing this user.

   vi. Add Update permission in addition to Read and Execute already required to access GAX.

b. Provide the user permission to modify the user's own configuration object:

   i. Select Configuration.

   ii. From the Accounts pane, choose Persons.

   iii. Click to open the required Person's properties.

iv. Select Permissions tab.
v. Add the Person or Access group containing this user.
vi. Add set Update permission.

# Pulse Configuration Options

The application templates for Pulse might contain other configuration options that are not described in this chapter. These options must remain set to the default values that are provided in the application templates. Changing these values might cause unexpected behavior.

You are not required to configure any options to start Pulse. Pulse supplies default values for all options that it requires to function.

## GAX Application Object

### [pulse] Section

**access_url**
Valid Values: http://<gax_host>:<gax_port>
where:
gax_host—The name or IP address of the GAX host computer
gax_port—The port of the GAX server
For example: http://pulsedev01:8280/gax
Default Value: http://host:port
Changes Take Effect: After restart
Specifies the URL of the Genesys Administrator Extension (GAX) in which Pulse has been configured. Configure this option only for High Availability (HA) configuration. You must configure this option for every GAX instance. For example, configure GAX1 host computer information for «access_url» GAX1 and GAX2 host computer information for «access_url» GAX2. If Pulse is behind a network load balancer, then both URLs should be empty.

Pulse shows the `Collector does not respond` error containing a link to the Pulse backup if available when the access_url option is configured and not empty.

Pulse shows the latest available data even if Collector is down, along with a `Collector does not respond` error when access_url option is empty.

**Note:** The URL must be accessible from the Pulse user network. If Pulse is behind a load balancer, then access_url should be empty. In scenarios with a load balancer, the health-check must be configured.

**cache_expire_timeout**
Valid Values: zero or any positive number
Default Value: 1200
Changes Take Effect: After restart
Specifies how long, in seconds, that Pulse stores the results of the object accessibility check in order to reduce the load on the Configuration server.

**editable_templates**
Valid Values: true,false

Default Value: false
Changes Take Effect: After restart
Enables users with appropriate permissions to edit Genesys-provided templates. If false, even and Administrator cannot edit the default templates. Use this option to remove obsolete or unused templates (for example, iWD or Email). Use in conjunction with the install_templates option.

**health_expire_timeout**
Valid Values: zero or any positive number
Default Value: 30
Changes Take Effect: After restart
Specifies how long, in seconds, Pulse stores result of previous health check, which includes the heartbeat, DB connection, and Configuration Server connection.

**install_templates**
Valid Values: true,false
Default Value: true
Changes Take Effect: After restart
Used in conjunction with the editable_templates option, specifies whether Pulse installs or updates the Genesys-provided templates for the current release when Pulse starts. In most cases, when editable_templates is true this option should be set to false.
For example, if you set editable_templates option to true, delete or edit some templates and the install_templates option is set to true, then Pulse restores all original Genesys-provided templates after restart for the current release. Pulse restores only templates from the current release, not earlier releases, which may be obsolete.

**snapshot_expire_timeout**
Valid Values: positive integer
Default Value: 24
Changes Take Effect: After restart
Specifies how long, in hours, that the snapshot file remains before Pulse automatically removes it. This setting should be no more than 24 hours, unless you plan to provide enough disk space to store additional snapshots.

# Pulse Collector Application Object

## [collector] Section

**hostname**
Valid Values: Valid host name
Default Value: Empty value
Changes Take Effect: After restart
Specifies the Simple Network Management Protocol (SNMP) host name.

**management-port**
Valid Values: Positive integers
Default Value: No default value

**Warning!** No other application should use this port.
Changes Take Effect: After restart
Specifies the TCP/IP port that Pulse Collector reserves for SNMP Option Management Client connections. If this option is absent or null, a server for Management Client is not created.
**Warning!** You must specify a value for this option if you are using an SNMP connection. Do not change the value for this option while Pulse Collector is running.

**output-transport**
Default Value: No default value (but the application template supplies the preconfigured transport name of gpb-out)
Valid Values: the names of all transport protocols that you have defined in the [transport-gpb-out] section.
Changes Take Effect: After restart
Specifies the output transport to be used to produce outputs.

## [configuration-monitoring] Section

**check-layout-presence-timeout**
Valid Values: 0-3600
Default Value: 900
Changes Take Effect: After restart
Specifies how often, in seconds, Pulse Collector checks for the deleted layouts. A zero (0) value completely disables the check.
**Note:** This defines the minimum timeout between two checks. The actual timeout depends on the database polling cycle, because the check is conducted after finishing subsequent database polling cycle.

**db-poll-period**
Valid Values: 3-3600
Default Value: 30
Changes Take Effect: After restart
Specifies how often, in seconds, Pulse Collector obtains updates from the Pulse database.
**Note:** Genesys recommends that you set this option to no less than 15 seconds.

**excluded-objects-propagation-delay**
Valid Values: 0...3600
Default value: 60
Changes Take Effect: Immediately
Specifies the delay in seconds before Pulse Collector attempts to propagate excluded objects in the affected layouts after an object is deleted. A zero value eliminates the timeout and Pulse Collector attempts to propagate excluded objects immediately after an object is deleted.

**metagroup-contents-recheck-delay**
Valid Values: 0...3600
Default Value: 60
Changes Take Effect: Immediately
Specifies the delay in seconds between when Pulse Collector verifies metagroup contents (such as Agent Group, Place Group, and DN Group) after notification from Configuration Server notifies Pulse Collector about changes in the contents of the metagroup object. Zero value of this option eliminates

timeout and metagroup change is processed immediately.
**Note:** This configuration option impacts ANY changes in the metagroup (for example, both adding and deleting objects), so new objects added to the metagroup appear in the layout after the specified delay.

**new-object-delay**
Valid Values: 0-86400
Default Value: 0
Changes Take Effect: Immediately
Specifies the delay, in seconds, between when Pulse Collector receives notification of a new object in Configuration Server, and when it starts to process this notification. Setting this option to 0 enables Pulse Collector to process new objects without delays.

**ods-wait-timeout**
Valid Values: 10–3600
Default Value: 300
Changes Take Effect: After restart
Specifies the time, in seconds, that Pulse Collector waits before re-checking the Pulse database for proper initialization.

**remove-dynamic-object-delay**
Valid Values: 0-600
Default Value: 60
Changes Take Effect: After restart
Specifies the time, in seconds, that Pulse Collector waits before excluding and closing statistics for the Agent from affected layout that was excluded from the StatServer-based VAG. A zero value turns off the delay and Pulse Collector processes changes immediately.

## [heartbeat] Section

**heartbeat-folder**
Valid Values: Valid folder path
Default Value: ./output/heartbeat (as provided by template file)
Changes Take Effect: After restart
Specifies the path in which Pulse Collector writes the heartbeat file.
**Note:** Relative file paths must begin with `./`

**heartbeat-period**
Valid Values: 3-3600
Default Value: 30
Changes Take Effect: After restart
Specifies the period of a heartbeat update, in seconds.

**heartbeat-success-condition**
Valid Values:
One or any combination of the following conditions:

- statserver—Stat Server connection should be available.

- snapshot-writer—The snapshot writer should be producing outputs without error.
- collector-db—The main DB connection should be available.

Default Value: statserver, snapshot-writer
Changes Take Effect: After restart
Specifies a comma-separated list of conditions to be checked for criteria of heartbeat success, which means the heartbeat is updated only if this criteria is met.

## [layout-validation] Section

**enable-layout-validation**
Valid Values: yes,no
Default Value: yes
Changes Take Effect: After restart
Enables or disables layout validation in Pulse Collector.
**Note:** Pulse Collector always checks for consistency of LayoutID and TenantID. These validations apply even if options are set to a value no.

**validate-strict-tenant-security**
Valid Values: yes,no
Default Value: yes
Changes Take Effect: After restart
Enables Pulse Collector to fail the layout validation when the option strict-tenant-security in the configuration-monitoring section is set to yes and Pulse encounters an individual object that belongs to a tenant that is different than layout's tenant.

## [limits] Section

**max-formulas-per-layout**
Valid Values: 1-100000
Default Value: 50
Changes Take Effect: After restart
Specifies the maximum number of formula-based statistics for each layout.

**max-metagroups-per-layout**
Valid Values: 1-100000
Default Value: 50
Changes Take Effect: After restart
Specifies the maximum number of metagroups for each layout.

**max-objects-per-layout**
Valid Values: 1-100000
Default Value: 400
Changes Take Effect: After restart
Specifies the maximum number of objects for each layout.

**max-statistics-per-layout**
Valid Values: 1-100000
Default Value: 100
Changes Take Effect: After restart
Specifies the maximum number of statistics for each layout.

## [localization] Section

**default-snapshot-message-language-code**
Valid Values: Language code (non-empty string, typically 2 letters)
Default Value: en
Changes Take Effect: After restart
Specifies the default language code to use for layout snapshot messages, if the layout snapshot does not override this value.

**language-pack-folder**
Valid Values: Folder path
Default Value: ./messages
Changes Take Effect: After restart
Specifies the folder from which Pulse Collector reads localized message files to show layout errors in the localized form.
**Note:** Relative file paths must begin with `./`

## [Log] Section

**all**
Valid Values:

- stdout—Log events are sent to the Standard output (stdout).
- stderr—Log events are sent to the Standard error output (stderr).
- network—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores log events in the Log Database.

  Setting the all log-level option to network enables Data Sourcer to send log events of Standard, Interaction, and Trace levels to Message Server. Log events of Debug level are neither sent to Message Server nor stored in the Log Database.
- memory—Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance.
- [filename]—Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application’s working directory.

Default Value: stdout
Changes Take Effect: Immediately
Specifies the outputs to which Data Sourcer sends all log events. You must separate log-output types with commas when you configure more than one output type.
For example, all = stdout, logfile
**Note:** To ease the troubleshooting process, consider using unique names for log files that different applications generate.

**buffering**
Default Value: false
Valid Values: true,false
Changes Take Effect: Immediately
Specifies whether the operating system file buffering is on or off. This option applies only to stderr and stdout output. Setting this option to true increases output performance.
**Note:** When you enable buffering, log messages might appear in the log after a delay.

**segment**
Valid Values:

- false—No segmentation allowed.
- <number> KB or <number>—Sets the maximum segment size in kilobytes. The minimum segment size is 100 KB.
- <number> MB—Sets the maximum segment size, in megabytes.
- <number> hr—Sets the number of hours for which the segment stays open. The minimum number is 1 hour.

Default Value: 10 MB
Changes Take Effect: Immediately
Specifies if there is a segmentation limit for a log file. If there is, this option sets the unit of measurement along with the maximum size. If the current log segment exceeds the size set by this option, the current file is closed and a new file is created.

**verbose**
Valid Values:

- all—All log events (that is, log events of Standard, Trace, Interaction, and Debug levels) are generated if you set the debug-level option in the statserver section to all.
- debug—The same as all.
- trace—Log events of the Trace and higher levels (that is, log events of Standard, Interaction, and Trace levels) are generated, while log events of the Debug level are not generated.
- interaction—Log events of the Interaction and higher levels (that is, log events of Standard and Interaction levels) are generated, while log events of the Trace and Debug levels are not generated.
- standard—Log events of the Standard level are generated, while log events of the Interaction, Trace, and Debug levels are not generated.
- none—Produces no output.

Default Value: all
Changes Take Effect: Immediately Determines whether a log output is created. If it is, this option specifies the minimum level of log events that are generated. The log-event levels, starting with the highest-priority level, are Standard, Interaction, Trace, and Debug.
Refer to the Framework Deployment Guide or Framework Solution Control Interface Help for more information on the Standard, Trace, Interaction, and Debug log levels.

## [object-name-format] Section

You can use a custom format for an object name, which can include a mix of predefined text and additions to the object properties within their actual values with optional width and trimming rules. For details, see “Valid object name format string” and “Object Information”.

**AccessResource**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Access Resource. For details, see “Object Properties” at the bottom of this section.

**ACDPosition**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype ACD Position. For details, see “Object Properties” at the bottom of this section.

**ACDQueue**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for the object of the of type DN with subtype ACD Queue. For details, see “Object Properties” at the bottom of this section.

**Agent**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for the object of the of type Agent. For details, see “Object Properties” at the bottom of this section.

**AgentGroup**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for the object of the of type Agent Group. For details, see “Object Properties” at the bottom of this section.

**CallingList**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Calling List. For details, see “Object Properties” at the bottom of this section.

**Campaign**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Campaign. For details, see “Object Properties” at the bottom of this section.

**CampaignCallingList**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Campaign Calling List. For details, see “Object Properties” at the bottom of this section.

**CampaignGroup**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Campaign Group. For details, see “Object Properties” at the bottom of this section.

**Cellular**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Cellular. For details, see “Object Properties” at the bottom of this section.

**Chat**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Chat. For details, see “Object Properties” at the bottom of this section.

**CoBrowse**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype CoBrowse. For details, see “Object Properties” at the bottom of this section.

**CP**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype CP. For details, see “Object Properties” at the bottom of this section.

**Data**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Data. For details, see “Object Properties” at the bottom of this section.

**DN**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN. For details, see “Object Properties” at the bottom of this section.

**DNGroup**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN Group. For details, see “Object Properties” at the bottom of this section.

**EAPort**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype EA Port. For details, see “Object Properties” at the bottom of this section.

**Email**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Email. For details, see “Object Properties” at the bottom of this section.

**Extension**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Extension. For details, see “Object Properties” at the bottom of this section.

**ExtRoutingPoint**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Ext Routing Point. For

details, see “Object Properties” at the bottom of this section.

**FAX**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Fax. For details, see “Object Properties” at the bottom of this section.

**Mixed**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Mixed. For details, see “Object Properties” at the bottom of this section.

**Music**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Music. For details, see “Object Properties” at the bottom of this section.

**Place**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Place. For details, see “Object Properties” at the bottom of this section.

**PlaceGroup**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Place Group. For details, see “Object Properties” at the bottom of this section.

**RoutingPoint**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Routing Point. For details, see “Object Properties” at the bottom of this section.

**RoutingQueue**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart

Specifies the object name formatting rule for an object of type DN with subtype Routing Queue. For details, see “Object Properties” at the bottom of this section.

**RoutingStrategy**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Routing Strategy. For details, see “Object Properties” at the bottom of this section.

**Script**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Script. For details, see “Object Properties” at the bottom of this section.

**ServiceNumber**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Service Number. For details, see “Object Properties” at the bottom of this section.

**StagingArea**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Script with subtype Staging Area. For details, see “Object Properties” at the bottom of this section.

**Switch**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Switch. For details, see “Object Properties” at the bottom of this section.

**Tenant**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Tenant. For details, see “Object Properties” at the bottom of this section.

**Video**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%

Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Video. For details, see “Object Properties” at the bottom of this section.

**VirtACDQueue**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Virt ACD Queue. For details, see “Object Properties” at the bottom of this section.

**VirtRoutingPoint**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Virt Routing Point. For details, see “Object Properties” at the bottom of this section.

**VoiceMail**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Voicemail. For details, see “Object Properties” at the bottom of this section.

**VoIP**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype VoIP.

**Workbin**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type Script with subtype Workbin. For details, see “Object Properties” at the bottom of this section.

**Workflow**
Valid Values: Valid object name format string. For details, see the table at the bottom of this section.
Default Value: %ObjectName%
Changes Take Effect: After restart
Specifies the object name formatting rule for an object of type DN with subtype Workflow For details, see “Object Properties” at the bottom of this section.

## Valid object name format string

The valid object name format string is free text string, which allows values of object properties: %<PropertyName>:<side><padding><length>%

- PropertyName—Specifies the property name.
  **Note:**Property names are case sensitive.
- side—Specifies the side, L (left) or R (right) , from where the length must be counted. If you do not specify a side, Pulse uses L by default.
  L is commonly used for string or text properties.
  R is commonly used for numbers.
- padding—Specifies the padding when the property value length is less than the specified custom length
  . (dot) to pad with space characters
  0 (zero) to pad with zero characters.
- length—Specifies the maximum number of characters for the property name.

| Valid Object Name Format String | Description |
|---|---|
| • %EmployeeID:10%<br>• %EmployeeID:L10% | Both specify 10 characters of property EmployeeID from the left side. |
| • %EmployeeID:.10%<br>• %EmployeeID:L.10% | Specifies 10 characters of property EmployeeID from left, but if length was less than 4 symbols, pad it with spaces. |
| • %DBID:R4% | Specifies 4 characters of property DBID from the right side. |
| • %DBID:R04% | Specifies 4 characters of property DBID from the right side, but if the length is less than 4 characters, pads it with zeros. |

## Object Properties

The table below lists the object properties available for use in the format strings.

| Object Type | Property | Description |
|---|---|---|
| All Object Types | DBID | Specifies the ID of the object. For example, Campaign Group is in group DBID, Campaign Calling List is in CallingList ID. |
| All Object Types | ObjectID | Specifies the ID number of the object in Pulse Collector.<br>**Notes:**<br>• This is the typically the configuration layer DBID, but |

| Object Type | Property | Description |
|---|---|---|
| | | for some types of objects (for example, Campaign Group, Campaign Calling List) this is a composite 64-bit ID.<br>• The composite 64-bit ID is an unsigned 64-bit number with the following composition:<br>• higher 32 bits: DBID of Campaign<br>• lower 32 bits: DBID of Calling List or Agent/Place group |
| All Object Types | ObjectName | Specifies the name of the object, which is written to the snapshot file. |
| All Object Types | ObjectType | Specifies the type of the object. |
| All Object Types | TenantID | Specifies the Tenant ID of the object. |
| All Object Types | type | Specifies the type of the object. |
| Agent | EmailAddress | Specifies the email address of the agent (person). |
| Agent | EmployeeID | Specifies the employee ID of the agent (person). |
| Agent | ExternalID | Specifies the external ID of the agent (person). |
| Agent | FirstName | Specifies the first name of the agent (person). |
| Agent | LastName | Specifies the last name of the agent (person). |
| Agent | UserName | Specifies the user name of the agent (person). |
| Calling List | Description | Describes the calling list. |
| Campaign | Description | Describes the campaign. |
| Campaign Calling List | CallingListDescription | Describes the underlying Calling List object. |
| Campaign Calling List | CallingListName | Specifies the DBID of the Calling List object. |
| Campaign Calling List | CampaignDBID | Specifies the DBID of the Campaign object. |
| Campaign Group | CampaignDBID | Specifies the DBID of the Campaign object. |
| Campaign Group | GroupDBID | Specifies the DBID of the group object. |

| Object Type | Property | Description |
|---|---|---|
| Campaign Group | GroupType | Specifies the numeric Type of the group object (CFGAgentGroup or CFGPlaceGroup). |
| DN, Routing Queue, Routing Point | Alias | Specifies the DN Alias. |
| DN, Routing Queue, Routing Point | AliasOrNumber | Populated with the DN Alias if available, otherwise populated with the DN number. |
| DN, Routing Queue, Routing Point | Number | Specifies the DN number. |
| DN, Routing Queue, Routing Point | SwitchDBID | Specifies the switch DBID. |
| DN, Routing Queue, Routing Point | SwitchID | Specifies the switch name. |
| Routing Strategy, Staging Area, Workbin | ScriptType | Specifies the script type ID. |
| Switch | DNRange | Specifies the switch DN Range. |
| Switch | SwitchType | Specifies the switch type. |

## [output] Section

**collector-snapshot-log-level**
Valid Values: Debug, Info, Warning, Error, Fatal, Unknown, None (case-insensitive)
Default Value: Warning
Changes Take Effect: After restart
Determines minimum log level for snapshot message that is written to Pulse Collector log.

**max-output-interval**
Default Value: 3600
Valid Values: 3–3600
Changes Take Effect: After restart
Specifies the maximum allowed output interval for all report layouts. Users can independently set output frequencies by layout within the Pulse user interface. If the set frequency, however, is greater than the value of this option, Pulse uses the value of this option instead.
**Note:** The value of this option must be greater than the value of the min-output-interval option or Pulse Collector logs an appropriate error.

**min-output-interval**
Default Value: 3
Valid Values: 3–3600
Changes Take Effect: After restart
Specifies the minimum allowed output interval for all report layouts. Users can independently set output frequencies by layout within the Pulse user interface. If the set frequency, however, is less than the value of this option, Pulse uses the value of this option instead.

**snapshot-log-level**
Valid Values: Debug, Info, Warning, Error, Fatal, Unknown, None (case-insensitive)
Default Value: Info
Changes Take Effect: After restart
Specifies the minimum log level for a snapshot message that is put to a layout snapshot.

## [parallel-processing] Section

**snapshot-builder-worker-thread-count**
Valid Values: 1-128
Default Value: 1
Changes Take Effect: After restart
Defines the number of concurrent threads used to build snapshots.

## [scripting] Section

**definition-script-execution-timeout**
Valid Values: 1-900
Default Value: 45
Changes Take Effect: After restart
Time in seconds allowed for definition script execution.

**formula-script-execution-timeout**
Valid Values: 1-900
Default Value: 5
Changes Take Effect: After restart
Time in seconds allowed for single formula evaluation script execution.

**init-script-execution-timeout**
Valid Values: 1-900
Default Value: 60
Changes Take Effect: After restart
Time in seconds allowed for initialization script execution.

**js-lib-path**
Valid Values: Valid folder paths
Default Value: ./jslib/standard
Changes Take Effect: After restart
Comma-separated list of locations of the directories that contain additional JavaScript libraries to be used within the formula scripting engine.
**Note:** Relative file paths must begin with `./`

**js-modules**
Valid Values: Comma-separated list of JavaScript files
Default Value: collector.js,cfglib.js,statlib.js,gts.js
Changes Take Effect: After restart
Comma-separated list of modules to preload into the scripting engine.

**stop-compute-formula-threshold-for-snapshot**
Valid Values: 0-100
Default Value: 3

Change Take Effect: After restart
Specifies the maximum allowed number of `timeout expired` errors during formula computation, after which, Pulse Collector assigns the particular formula-based statistic the `ERROR` value for the current snapshot. A zero value of this option suppresses the limit of the `timeout expired` failures.

## [statistic-request-handling] Section

**always-use-statserver-newapi**
Valid Values: yes, no
Default Value: no
Changes Take Effect: After restart
Determines whether to force Pulse Collector to request all statistics through the StatServer New API that uses the proper parameter set.

**data-flow-check-interval**
Default Value: 360
Valid Values: 5–3600
Changes Take Effect: After restart
Recommended Values: Positive integers that are less than the maximum notification interval value, either with reset-based notification using the time profile or time-based notification using the notification interval parameter.
**Note:** Standard Pulse layouts use a 60 second notification interval.
Changes Take Effect: Immediately
How often, in seconds, Pulse Collector verifies that there is data flow from Stat Server (primary and backup), provided that Pulse Collector contains active report layouts and has a live connection to Stat Server.

**data-flow-timeout**
Default Value: 120
Valid Values: 5–86400
Changes Take Effect: After restart
Recommended Values: Positive integers that are greater than the value that you specify for data-flow-check-interval and less than two times of the maximum notification interval value, either with reset-based notification using time profile or time-based notification using notification interval parameter.
**Note:** Standard Pulse layouts use a 60 second notification interval.
Changes Take Effect: Immediately
Time, in seconds, that is acceptable to suspend the transfer of data from Stat Server (primary or backup), provided that Pulse Collector contains active report layouts and has a live connection to Stat Server. The timer starts upon the last successful data transfer.

**data-source-choice-strategy**
Valid Values: PrimaryRunMode, LastGood, FirstAvailable, MostUpToDate, PrimaryInCME
**Note:** All values are case-insensitive.
Default Value: PrimaryRunMode
Changes Take Effect: After restart
Specifies which StatServer configured in the configuration layer that Pulse Collector uses as a data source for a snapshot:

- PrimaryRunMode—Pulse Collector uses the StatServer running in the primary mode. If both Stat Servers appear to be backup, Pulse Collector attempts the next strategy.

- LastGood—Pulse Collector uses the last good StatServer if available. Otherwise, Pulse Collector attempts the next strategy.
- FirstAvailable—Pulse Collector uses the Primary StatServer if available. If uses the Primary StatServer is unavailable, Pulse Collector uses the Backup StatServer. Otherwise, Pulse Collector attempts the next strategy.
- MostUpToDate—Pulse Collector uses StatServer that sent statistic data with most recent Server Time. Otherwise, Pulse Collector attempts the next strategy.
- PrimaryInCME—Pulse Collector always uses the Primary StatServer.

**statserver-batch-size**
Valid Values: 1-10000
Default Value: 500
Changes Take Effect: Immediately
Specifies the number of statistic requests sent to Stat Server in a single packet. The recommended value depends on the number of statistic requests you plan to run, network bandwidth, and processing capabilities of the Stat Server and Pulse Collector servers. If Stat Server disconnects Pulse Collector when it is opening statistics with error message Client too slow, decrease value of this option.

**statserver-profiles-timeout**
Valid Values: 1-86400
Default Value: 600
Changes Take Effect: Immediately
Specifies the timeout, in seconds, to receive and process server profiles from Stat Server. If profiles are not received and processed within the given timeout, Pulse Collector closes the current connection to StatServer and attempts to reconnect.

**suspend-statistic-notifications-for-paused-layouts**
Valid Values: yes, no
Default Value: no
Changes Take Effect: After restart
Determines whether to suspend the statistic notifications for paused layouts.
**Note:** You must have Stat Server version 8.1.200.17 or higher for this functionality to work correctly, if you set the value of suspend-statistic-notifications-for-paused-layouts to yes.

**verbose-request-statistics**
Valid Values: true, false
Default Value: false
Changes Take Effect: Immediately
Determines whether to enable Pulse Collector to log verbose messages about the objects for which it requests statistics.

## [transport-gpb-out] Section

**output-file-ext**
Valid Values: Valid file extensions for your operating system
Default Value: gpb

Changes Take Effect: After restart
Specifies the file extension for the full output file format.

**output-file-mode**
Valid Values: 0-0777
Default Value: 0664
Changes Take Effect: After restart
On Linux, specifies the UNIX mode for each output file created by this transport.
**Important!** This option respects umask set at OS level.

**output-folder**
Valid Values: Valid folder paths
Default Value: No default (the collector.apd file supplies the value of a sample output directory)
Changes Take Effect: After restart
Specifies the path in which Pulse Collector writes output files. If you specify a folder that does not exist, Pulse Collector creates it for you.
**Note:** Relative file paths must begin with `./`

**transport-type**
Default Value: file
Valid Values: file
Changes Take Effect: After restart
Specifies the transport output type.

# Starting and Stopping Pulse

Pulse starts and stops automatically when Genesys Administrator Extension (GAX) starts and stops.

## Prerequisites

Confirm the following requirements before starting Pulse:

a. See Browser Support for Pulse in the *Genesys Supported Operating Environment Reference Guide*.
b. Your monitor must be set to a resolution of no less than 1024x768.
c. The DB Server for Pulse Collector must be running.
d. The RDBMS for the Pulse database must be running.
e. Configuration Server and the DB Server for Configuration Server must be running.

> **Important**
>
> Users without permission to edit their own Person object cannot save their Custom dashboard configuration. Users without permission to edit both their own Person object and their Tenant object cannot save the Default dashboard configuration.
>
> For example, such a user can make changes, such as creating widgets, but cannot save the changes. The next time the page is loaded, the changes are lost.
>
> See the steps to configure user access for details.

# Starting and Stopping Pulse Collector

You can start Pulse Collector on a Windows or UNIX platform. Invoking Pulse Collector starts a series of internal checks for proper configuration. The value of the `max-output-interval` option, for example must be greater than the value of the `min-output-interval` option or Pulse Collector exits. Verify your Collector Application object for proper configuration before starting Pulse Collector.

## Prerequisites

The following must be running prior to starting Pulse Collector:

- Backup Configuration Server

  To restart the Pulse Collector application when the backup Configuration Server switches to the primary mode, you must specify the backup Configuration Server parameters when starting Pulse Collector.

  On the command line, specify these parameters using the following two arguments:

  - -backup_host hostname
  - -backup_port port-number

- National Characters

  To use national characters in the string properties correctly, Pulse Collector determines which multibyte encoding is used by Configuration Server. You can allow Pulse Collector to use the default setting or specify the characters by editing the following configuration options:

  - **Windows**

    You can specify Configuration Server multibyte encoding through the command-line parameters:

    - `-cs_codepage` following the Windows code page number (for example, 1251). For more information about Windows code pages and their numbers, see http://msdn.microsoft.com/en-us/library/windows/desktop/dd317756(v=vs.85).aspx If the parameter is not specified, Pulse Collector assumes Configuration Server uses the code page that corresponds to the locale of the Windows operating system where Pulse Collector is running.
    - `-cs_encoding`. It accepts Microsoft-compatible symbolic name of locale and results choice of default code page for that locale. For more information about Microsoft-compatible locale names, see http://msdn.microsoft.com/en-us/library/dd373814.aspx

    > Important
    >
    > `{{{1}}}`

  - **Linux**

    You can specify Configuration Server multibyte encoding through the command-line parameter `-cs_encoding` following the iconv-compatible symbolic name of encoding (for example, UTF-8).

To display the list of encodings supported on the your system, enter the following command in the Linux console:

```
iconv --list
```

# Start Pulse Collector

## Starting Pulse Collector

You can start Pulse Collector on any of the supported platforms.

### Solution Control Interface (SCI)

You can start Pulse Collector:

**[+] From SCI.**

To start Pulse Collector from the Solution Control Interface (SCI):

1. From the `Applications` view, select your Pulse Collector `Application` object in the list pane.
2. Click the `Start` button on the toolbar, or select `Start` either from the Action menu or the context menu. (Right-clicking your `Application` object displays the context menu.)
3. Click `Yes` in the confirmation box that appears. Your Pulse Collector application starts.

For information about how to use SCI, refer to *Framework 8.0 Solution Control Interface Help.*

### Windows

On a Windows platform, you can start Pulse Collector:

**[+] Manually from the Programs menu as an application.**

To start Pulse Collector from the `Programs` menu as an application, select `Start Pulse Collector` from the program group created during installation. The application opens a console window and automatically issues the parameters specified during configuration to start Pulse Collector. The Pulse Collector application name, version, and connectivity parameters appear in the title bar.

**[+] Manually from a console window as an application.**

1. To start Pulse Collector as an application from a console window: At the command-line prompt, go to the directory where Pulse Collector has been installed. Type the name of the Pulse Collector executable followed by the appropriate command-line parameters using the following syntax:

   ```
   pulse_collector.exe -host hostname -port portno -app application
   ```

   where:

   - `hostname` refers to the name of the computer on which Configuration Server is running.
   - `portno` refers to the communication port on which Configuration Server is running.

- `application` refers to the name of the Pulse Collector `Application` object as defined in Genesys Administrator.

> Important
>
> If the host or application name contains spaces or hyphens (–), enclose it in double quotation marks.

For example, to start Pulse Collector with parameters specifying the host as cs-host, port as 2020, and name as Pulse Coll, type:

```
pulse_collector.exe -host "cs-host" -port 2020 -app "Pulse Coll"
```

> Important
>
> If needed, specify the optional parameters -backup_host, -backup_port, -cs_codepage, and -cs_encoding.

**[+] As a Windows service.**

1. From the task bar, choose `Start - Administrative Tools > Computer Management`.
2. Open `Services and Applications > Services`.
3. Right-click your Pulse Collector service from the list and select `Start`.

> Important
>
> Since the Local Control Agent (LCA) can be installed as a Windows service with the user interface disabled, all servers started through SCI, in this case, are started without a console unless you specifically select the Allow Service to Interact with Desktop check box for both LCA and Pulse Collector.

### UNIX Platforms

You can start Pulse Collector:

**[+] Manually from UNIX Platforms.**

1. Go to the directory where Pulse Collector has been installed.

> **Important**
>
> You can invoke Pulse Collector only from the directory in which it was installed.

2. Type the name of the Pulse Collector executable followed by the appropriate command-line parameters using the following syntax:
   ```
   ./pulse_collector -host hostname -port portno -app application
   ```

   where:

   - `hostname` refers to the name of the computer on which Configuration Server is running.
   - `portno` refers to the communication port on which Configuration Server is running.
   - `application` refers to the name of the Pulse Collector Application object as defined within Genesys Administrator.

> **Important**
>
> If the host or application name contains spaces or hyphens (–), enclose it in double quotation marks.

For example, to start Pulse Collector with parameters specifying the host as `cs-host`, port as `2020`, and name as `Pulse Coll`, type:
```
./pulse_collector -host "cs-host" -port 2020 -app "Pulse Coll"
```

> **Important**
>
> If needed, specify the optional parameters -backup_host, -backup_port, and -cs_encoding.

Configure the `stdout` option in the `log` section of `collector` options to write to a log file, so that you can check for errors in its configuration if Pulse Collector fails to start. If you cannot resolve a problem, contact Genesys Customer Care and provide the entire content of the log.

You can also type the name of the Pulse Collector executable and its command-line parameters into a shell script and execute the script using the following command:
```
./run.sh [Name of script]
```

To redirect Pulse Collector output (on most UNIX shells), use the following syntax:
```
./pulse_collector -host hostname -port portno -app appl > log_file.log
```

To have both log file and console, within Genesys Administrator add the following to Pulse Collector’s application properties:

- Section: `Log`
- Option: `all` with the following value:
  `stdout,<log_file_name.log>,network`
  Separate values with commas. Instead of `stdout`, you can also specify `stderr`.

  Pulse Collector writes messages to <log_file_name.log> in the same directory where Pulse Collector is installed. To have Pulse Collector write to a different location, specify the full path for this parameter.

### End of procedure

## Stop Pulse Collector

## Stopping Pulse Collector

You can stop Pulse Collector on any of the supported platforms.

> **Important**
>
> Be sure that the autorestart property is cleared for the Pulse Collector Application object in Genesys Administrator.

### Solution Control Interface (SCI)

You can stop Pulse Collector:

**[+] From SCI.**

If you use LCA and Solution Control Server (SCS), you can stop Pulse Collector from SCI. To do so:

1. From the `Applications` view, select your Pulse Collector `Application` object in the list pane.
2. Click `Stop` on the toolbar, or select `Stop` either from the Action menu or the context menu.
3. Click Yes in the confirmation box that appears.

### On a Windows platform

On a Windows platform, you can stop Pulse Collector:

- **[+] Manually from its console window as an application.**

  If Pulse Collector is running as an application, switch to its console window and press `Control-C` to stop it.

- **[+] As a Windows service.**

If you are running Pulse Collector as a Windows service, you should stop it only from the Microsoft Management Console.

To stop Pulse Collector running as a Windows service:

1. From the task bar, choose `Start - Administrative Tools > Computer Management`.
2. Open Services and `Applications > Services`.
3. Right-click your Pulse Collector service from the list and select `Stop`.

- If LCA and SCS are used, you can stop Pulse Collector from SCI.

## On a Unix platform

Stop Pulse Collector on UNIX using either of the following methods:

- On the command line, type `kill -9 processid`, where `processid` is Pulse Collector’s UNIX process ID.
- Press `^C` from the active Pulse Collector console.
- If LCA and SCS are used, you can stop Pulse Collector from SCI.

# Application Files

The Pulse installation routine creates a root application folder with a scripts subfolders.

## Pulse

The following files are found in the Pulse root folder and the scripts subfolder.

**[+] Contents of the Pulse Root Folder**

- ip_description.xml—File that stores the installation package content.
- protobuf-java-2.5.0.jar—Jar file that contains the Google Protocol Buffers library.
- pulse.jar—Executable Jar file.
- pulse-proto—Executable Jar file.
- read_me.html file—File that contains general information about this product release.
- templates.xls—File that contains template descriptions.
- dbtool—Folder that contains dbtool files.
- scripts subfolder—Subfolder containing:
    - Subfolders that hold the database initialization SQL scripts for each of the following RDBMS types:
        - Microsoft SQL
        - Oracle
        - PostgreSQL
    - pulse_statistics.cfg file—File that specifies the options needed for Stat Server to process Pulse preconfigured templates.
- thirdparty—Folder that contains third-party licenses and additional files as required.

**[+] Contents of the scripts\[dbtype] Subfolder**

- mssql\pulse_init_mssql.sql—Database initialization script for a Microsoft SQL RDBMS.
- oracle\pulse_init_oracle.sql—Database initialization script for an Oracle RDBMS.
- postgres\pulse_init_postgres.sql—Database initialization script for a PostgreSQL RDBMS.

## Pulse Collector

Pulse Collector installation routine deploys the following files to the Collector root folder and its subfolders.

### [+] Contents of the Pulse Collector Root Folder

- common.lms—File that stores common log messages.
- collector and collector.exe—Application executables for Unix and Microsoft Windows platforms.
- collector.lms—File that stores Pulse Collector–specific log messages.
- extras—Folder that stores file extra configuration options.
- ip_description.xml—File that stores installation package content.
- messages—Folder that stores localized error messages.
- pthreadVC2.dll—The pthreads-win32 library file.
- pthreadVC2.pdb—File that provides pthreads-win32 debugging information.
- run.sh—On UNIX platforms, shell script that launches Pulse Collector.
- startServer.bat—On Windows platforms, the batch file that starts Pulse Collector as a regular application, not as a Windows Service.
- thirdparty—Folder that contains third party licenses and additional files as required.
- jslib—Folder that contains two subfolders, extras and standard, which contain the JavaScript library files to be used by Pulse Collector.

### [+] Contents of the Pulse Collector extras Folder

- finetuning.cfg—File that provides fine tuning configuration options.

### [+] Contents of the Pulse Collector jslib/standard Folder

- cfglib.js—File that provides JavaScript definitions related to Configuration Server
- collector.js—File that provides JavaScript definitions for Pulse Collector.
- gts.js—File that provides set of additional JavaScript functions used in the standard Pulse layout templates
- statlib.js—File that provides JavaScript definitions for various constants on which Stat Server relies.

### [+] Contents of the Pulse Collector messages Folder

- layout-en.msg—Localized message file.
- layout-ru.msg—Localized message file.
- layout-uk.msg—Localized message file.

# Change History

This section lists content that is new or that has changed significantly since the first release of this content. The most recent changes appear first.

For *New in This Release* content, see the Pulse landing page for this release.

## New in Document Version 8.5.0

The following topics have been added or changed since the previous release:

- The Architecture diagram was updated.
- Software requirements and deployments were updated in Deploying Pulse Collector and Pulse.
- The step to execute `ALTER DATABASE <Pulse DB> SET READ_COMMITTED_SNAPSHOT ON;` was added to Deploying the Pulse database for the Microsoft SQL Server.
- Starting Pulse was updated.
- Prerequisites and Starting Pulse were updated.
- The following Configuration Options were changed:
    - Removed the [`layout-history`], [`parallel-processing`], [`thresholds`], and [`transport-gpb-history`] sections.
    - In the [`collector`] section:
        - Removed the `history-transport` option.
    - In the [`configuration-monitoring`] section:
        - Added the `excluded-objects-propagation-delay` and `remove-dynamic-object-delay` options.
        - Removed the `auto-detect-dynamic-metagroups`, `enable-differential-layout-recheck`, `recheck-active-layout-definition`, and `strict-tenant-security` options.
    - In the [`layout-validation`] section:
        - Renamed the `enable-strict-tenant-security-validation-rule` option to `validate-strict-tenant-security`.
    - In the [`limits`] section:
        - Removed the `max-custom-actions-per-layout`, `max-custom-thresholds-per-layout`, `max-threshold-associations-per-layout` options.
    - In the [`output`] section:
        - Added the `collector-snapshot-log-level` and `snapshot-log-level` option.
        - Removed the `cleanup-output` option.
    - In the [`pulse`] section:

    - Added the editable_templates, install_templates, and snapshot_expire_timeout options.
    - Removed the pause_timeout option.
- In the [scripting] section:
    - Added the stop-compute-formula-threshold-for-snapshot option.
    - Removed the cumulative-formula-script-execution-timeout, cumulative-threshold-script-execution-timeout, enable-extended-statistic-value, post-processing-script-execution-timeout, and threshold-script-execution-timeout options.
- In the [statistics-request-handling] section:
    - Removed the optimize-statistic-requests options.
- In the [transport-gpb-out] section:
    - Added the output-file-mode option.
    - Removed the backend-file-ext, cleanup-file-modification-age-threshold, generate-layout-info-file, and output-format options.