



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 Help

Report Formulas

# Report Formulas

If you decide that one of your reports needs a different or additional statistic, you can edit the report's template to make that happen. You can accomplish this by adding a formula to the report template that retrieves the statistic or key performance indicator (KPI) you want.

Since you cannot change the standard templates provided, if you want to change one of the standard reports, just create a clone of the template and make changes in the new template.

Who can create these statistics? If you can create and edit Genesys Pulse templates, you can use formulas.

## Contents

- [1 Report Formulas](#)
  - [1.1 Add a formula](#)
  - [1.2 Display percentages](#)
  - [1.3 Display Agent Status KPIs](#)
  - [1.4 Display interaction properties](#)
  - [1.5 What do I do next?](#)

### Important

If you already know how to use the formulas, you can use [the function library](#) to help you create your formulas.

## Add a formula

The screenshot shows the 'Statistics' configuration interface. On the left, a list of statistics is displayed, with 'Answers' selected and highlighted in blue. The 'Add' button is visible at the top right of the list. The main configuration area on the right contains the following fields:

- Display Name \***: A text input field containing 'Answers'.
- Description**: A text area containing the text: 'The total number of dialing attempts initiated by a Campaign Manager with a call result of Answer (when a call is answered by a human voice). In some contact centers, the call result can also mean Right'.
- Alias \***: A text input field containing 'Campaign\_Answers'.
- Display Format \***: A dropdown menu set to 'Integer'.
- Formula**: A checkbox labeled 'Formula' is checked and circled with a black oval.
- Hide Statistic**: An unchecked checkbox.
- Show Agent State Icon**: An unchecked checkbox.

A 'Save' button is located at the bottom right of the configuration area.

From the statistic detail pane while editing a widget or template, you can create or customize statistics by creating a formula.

The formula uses a javascript-based syntax, which lets you calculate expressions with values given by other statistic and use functions provided by Genesys for more specific calculations. For example, you can calculate the ratio of the calls abandoned to the calls offered in your queue to measure the percentage of abandoned calls in your queue.

## Display percentages

**Statistics \*** Add

Hit Ratio		
Estimated Time		
Records Completed		
Dialed Abandoned		
Dialed Answering Ma...		
<b>Answers</b>		
Attempt Busies		
Attempts Cancelled		
Attempts made		
DoNotCall Results		
Dropped Results		
Fax Modem Results		
No Answer Result		

**Display Name \***  
Answers

**Description**  
The total number of dialing attempts initiated by a Campaign Manager with a call result of Answer (when a call is answered by a human voice). In some contact centers, the call result can also mean Right

**Alias \***  
Campaign\_Answers

**Display Format \***  
Integer

☒ **Formula**

☐ Hide Statistic ☐ Show Agent State Icon

Save

Let's say you want to display percentages based on two metrics. Just copy the following example using the statistics you want.

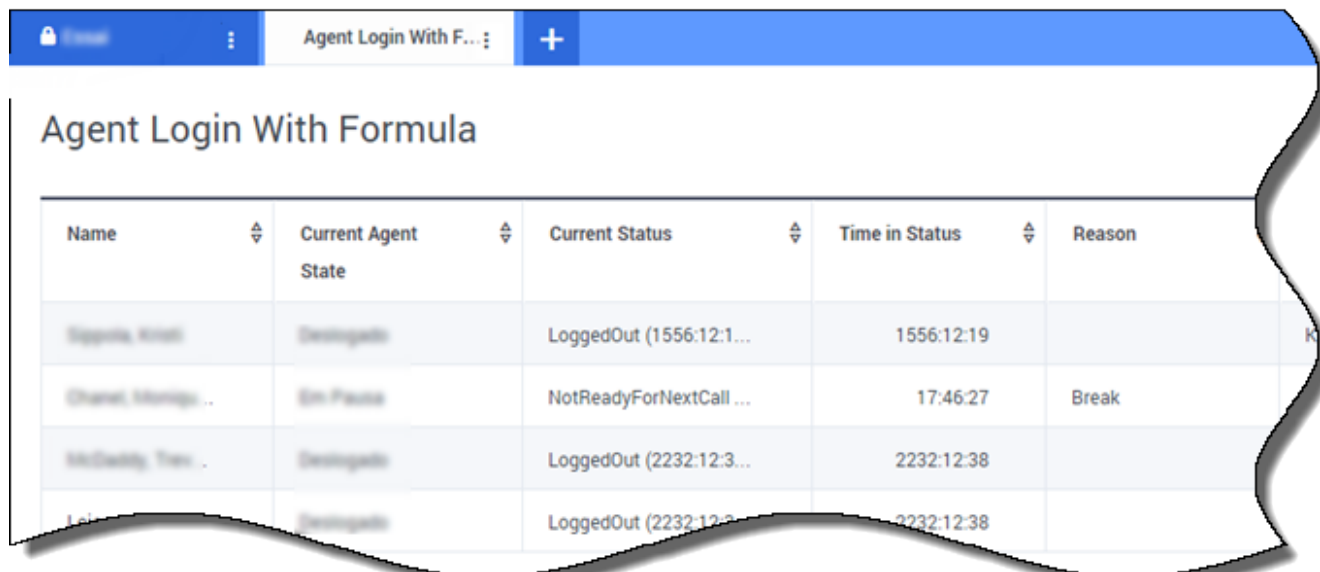
In this example, we want to retrieve the percentage of outbound calls out of the total of both inbound and outbound calls. The formula can access any statistic within a template with the following syntax: `Data.Statistic-Alias.Value`. The formula must return a valid Result value.

In the following formula, we assume the outbound calls are defined by a statistic alias Outbound and the inbound calls are Inbound.

### Formula: Calculate a Percentage

```
if ((Data.Outbound.Value + Data.Inbound.Value) != 0)
Result = 100 * Data.Outbound.Value / (Data.Outbound.Value + Data.Inbound.Value);
else Result = 0;
```

## Display Agent Status KPIs



The screenshot shows a report interface with a blue header bar containing a search icon, a filter icon, and the text 'Agent Login With F...'. Below the header, the report title 'Agent Login With Formula' is displayed. The main content is a table with the following data:

Name	Current Agent State	Current Status	Time in Status	Reason
Supra, Kristi	Disconnected	LoggedOut (1556:12:1...	1556:12:19	
Chanel, Monique ...	On Pause	NotReadyForNextCall ...	17:46:27	Break
McQuay, Tim ...	Disconnected	LoggedOut (2232:12:3...	2232:12:38	
Le...	Disconnected	LoggedOut (2232:12:3...	2232:12:38	

Let's say you want to display KPIs for agent status. Just use the `Current_Status` statistic.

### [+] How the `Current_Status` statistic is defined.

The `Current_Status` statistic is defined by Stat Server options properties. The statistic type `ExtendedCurrentStatus` returns a specific object that can be further analyzed to provide only the Duration of the object.

```
[ExtendedCurrentStatus]
Category=CurrentState
MainMask=*
Objects=Agent
Subject=DNAAction
```

You can use formulas to find the information you need:

### [+] Show agent time in current state

You can display the agent status duration using the `Current_Status` statistic.

**Formula: Get Status Duration**

```
Result = G.GetStatusDuration(Data.Current_Status.Value);
```

### [+] Show the Reason Code selected by the agent

You can display the reason code for the agent status.

#### Formula: Get Reason Code

```
Result = G.GetReasonCodes(Data.Current_Status.Value);
```

If you want to display more user data in addition to the Reason Code, you need to enable the Additional Data property (User Data) of the statistic and apply a formula to filter only the Reason Code from the resulting Current\_Status, which contains both the User Data and Reason code.

#### Formula: Filter only Reason Code

```
var res = G.GetReasonCodes(Data.Current_Status.Value);
var x = res.split(';');
Result = "";
for (var i = 0; i < x.length; i++) {
    var s = x[i];
    if (s.indexOf("Break") > -1 ||
        s.indexOf("Offline") > -1 ||
        s.indexOf("Training") > -1 ) { Result = s; break; }
}
```

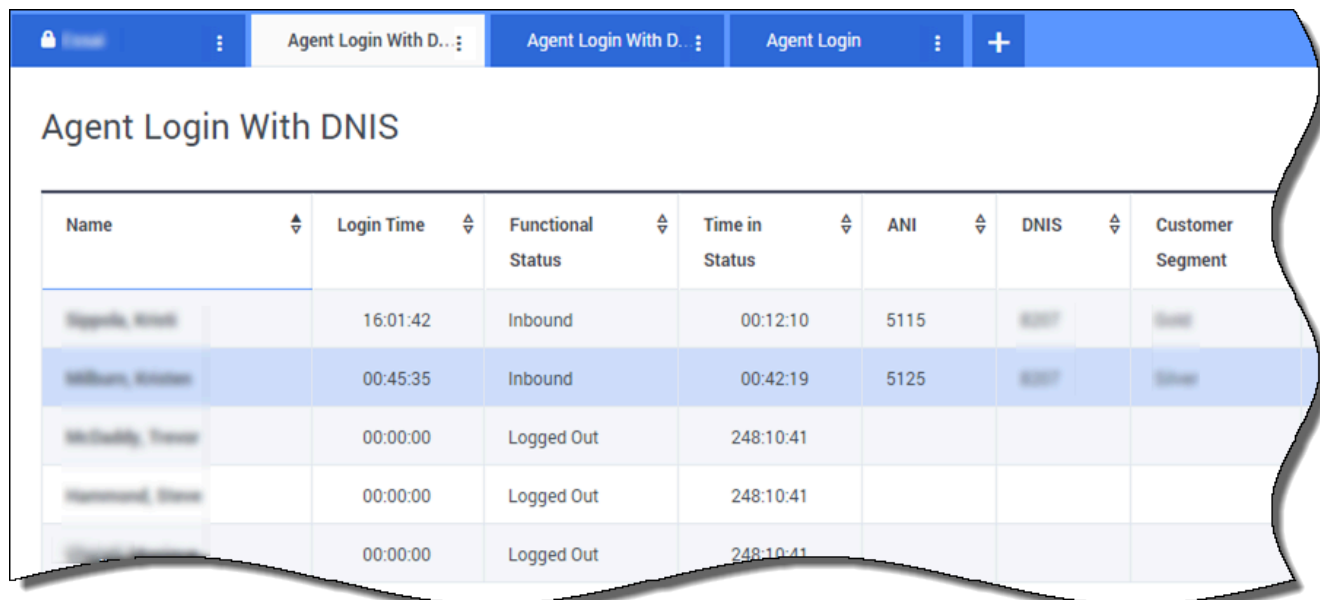
### [+] Show current agent state by media type

You can display the current agent state by media type.

**Formula - Get agent state by media type**

```
Result = G.GetAgentNonVoiceStatus(Data.Current_Status.Value,
'email');
```

## Display interaction properties



The screenshot shows a report interface with a blue header bar containing navigation icons and tabs. The active tab is 'Agent Login With DNIS'. Below the header, the report title 'Agent Login With DNIS' is displayed. The main content is a table with the following columns: Name, Login Time, Functional Status, Time in Status, ANI, DNIS, and Customer Segment. The table contains five rows of data, with the second row highlighted in blue.

Name	Login Time	Functional Status	Time in Status	ANI	DNIS	Customer Segment
Agarwal, Anil	16:01:42	Inbound	00:12:10	5115	8007	Gold
Williams, Andrew	00:45:35	Inbound	00:42:19	5125	8007	Silver
McQuibb, Trevor	00:00:00	Logged Out	248:10:41			
Hammond, Steve	00:00:00	Logged Out	248:10:41			
Smith, Andrew	00:00:00	Logged Out	248:10:41			

Let's say you want to display interaction properties including flow segmentation, ANI, and DNIS. You can use formulas to find the information you need:

**[+] Show the customer segment of the interaction**

You can display the customer segment defined by the CustomerSegment key-value pair of the interaction by using the following formula.

**Formula: Get Customer Segment**

```
Result = G.GetCustomerSegment(Data.Current_Status.Value);
```

### [+] Show the ANI of the customer

You can display the ANI of the customer by using the following formula.

#### **Formula: Get ANI**

```
[Result = G.GetANI(Data.Current_Status.Value);
```

### [+] Show the DNIS of the customer

You can display the DNIS of the customer by using the following formula.

#### **Formula: Get DNIS**

```
Result = G.GetDNIS(Data.Current_Status.Value);
```

## What do I do next?

You might want to learn more about:

- [Widget templates](#)
- [Statistic properties](#)
- [Template function library](#)