



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.

# Stat Server User's Guide

Statistical Type Sections

12/15/2025

# Statistical Type Sections

A *statistical type* (stat type) is a collection of actions, object types, category, and one subject that all help define the structure of a metric. Other factors may contribute to a metric's definition, such as a time profile, an optional time range, and an optional filter, all described earlier in this chapter.

Each stat type definition consists of:

- A user-defined section name, which represents the name of the stat type.
- Configuration options that apply to that section.

Most stat type configuration options can be classified as one of the following:

- Options for core stat types
- Options for Java stat types
- Options for Formula stat types

## Warning

The **Formula** category (as a value of the Category option) and the stat type definition option Formula (described in the table below) are two distinct objects.

A small number of the options serve both core and Java stat type classifications, but these options have differing permissible values. The table below lists all configuration options that you can use to define stat types. The third column in the table indicates that you can specify this option for stat types that are used in conjunction with a Stat Server Java extension.

Statistics that are based on core stat type definitions are calculated directly within Stat Server. Statistical values of Java stat types, on the other hand, are provided to Stat Server by another Genesys server, such as Interaction Server, Orchestration Server, or Outbound Contact Server.

## Stat Type Configuration Options

| Option   | Description   | Java |
|----------|---|------|
| Objects  | <p>Specifies a list of comma-separated Stat Server object types to which statistics apply. The list must consist of objects of the same compatibility group. You must include this option in a stat type definition and specify a value.</p> <p>Default Value: No default value</p> <p>Valid Values: Refer to <a href="#">Stat Server Object Types and Descriptions</a> and <a href="#">Campaign Objects</a>.</p> <p>Changes Take Effect: When Stat Server restarts.</p>  | Yes  |
| MainMask | <p>Specifies a list of comma-separated actions (or statuses in case of status-based Subject option in the Stat Type) that indicate which contact center events will be measured. This list comprises members from the following groups:</p> <ul style="list-style-type: none"><li>• regular DN actions</li><li>• mediation DN actions</li><li>• media-channel actions</li><li>• campaign actions</li><li>• statuses</li></ul> <p>This option is mandatory for core stat types and you must specify one or more values.</p> <p>Use the wildcard (*) character to specify all actions; use the logical NOT (~) character to exclude the action it precedes. Use parentheses around each action (or status) that you want Stat Server to exclude from consideration of being filtered. You cannot, however, use parentheses in conjunction with * or ~. For example:</p> <p>MainMask=CallInbound, (CallOutbound)</p> <p>Please note that the logical NOT (~) character does not work</p> |      |

| Option  | Description  | Java |
|---------|--|------|
|         | <p>with aliases for the group of actions, such as CallWait on mediation DN's. To exclude the whole group, each member should be excluded explicitly. For example:</p> <p><code>*,~CallWaitUnknown,~CallWaitConsult,~CallWaitInbound,~CallWaitOutbound</code></p> <p>If a filter were applied to a statistic having this MainMask designation, Stat Server would only apply the filter to CallInbound actions. CallOutbound actions would continue to contribute to the tally of this statistic unfiltered. It is also possible to use the * and ~ characters in selective filtering.</p> <p>Default Value: No default value</p> <p>Valid Values: Refer to <a href="#">Stat Server Actions</a>, <a href="#">Object Statuses</a>, and <a href="#">Campaign Operational Actions</a> for a listing and description of these actions and statuses.</p> <p>Changes Take Effect: When Stat Server restarts.</p> |      |
| RelMask | <p>Specifies a list of comma-separated actions (or statuses in case of status-based Subject option in the Stat Type) that indicate the superset of contact center events against which the listing of actions (or statuses) provided in the main mask will be measured. This list comprises members from one of the following groups:</p> <ul style="list-style-type: none"><li>• regular DN actions</li><li>• mediation DN actions</li><li>• media-channel actions</li><li>• campaign actions</li><li>• statuses</li></ul> <p>Specifying this option is not mandatory, but if you do use it, you must supply one or more values.</p> <p>Use the wildcard (*) character to specify all actions; use the</p>  |      |

| Option          | Description  | Java |
|-----------------|--|------|
|                 | <p>logical NOT (~) character to exclude the action it precedes; and, use parentheses around each mask that you want Stat Server to exclude from consideration of being filtered. You cannot, however, use parentheses in conjunction with * or ~.</p> <p>Please note that the logical NOT (~) character does not work with aliases for the group of actions, such as CallWait on mediation DNs. To exclude the whole group, each member should be excluded explicitly. For example:</p> <p><code>*,~CallWaitUnknown,~CallWaitConsult,~CallWaitInbound,~CallWaitOutbound</code></p> <p>Default Value: No default value</p> <p>Valid Values: Refer to <a href="#">Stat Server Actions</a>, <a href="#">Object Statuses</a>, and <a href="#">Campaign Operational Actions</a> for a listing and description of these actions and statuses.</p> <p>Changes Take Effect: When Stat Server restarts.</p> |      |
| Category        | <p>Informs Stat Server how to calculate statistics. This section is mandatory for both core and Java stat types. You must supply one and only one value.</p> <p>Default Value: No default value</p> <p>Valid Values:</p> <ul style="list-style-type: none"> <li>For Stat Server core stat types, refer to <a href="#">Statistical Categories</a>.</li> <li>For Java stat types, this value must be: <code>JavaCategory</code></li> </ul> <p>Changes Take Effect: When Stat Server restarts.</p>  | Yes  |
| JavaSubCategory | <p>The name of the Java subclass that implements statistic calculation.</p> <p>Default Value: No default value</p>   | Yes  |

| Option               | Description  | Java |
|----------------------|--|------|
|                      | <p>Valid Values: String specified in the following format:<br/><b>jarfile:subclass</b></p> <p>Changes Take Effect: When Stat Server restarts.</p>  |      |
| Subject              | <p>Specifies the subject type for statistics calculation that, when changed, affects the statistical value. This section is mandatory for core stat types and you must supply one and only one value.</p> <p>Default Value: No default value</p> <p>Valid Values:</p> <ul style="list-style-type: none"> <li>• DNAction, DNStatus, AgentStatus, GroupStatus</li> <li>• PlaceStatus, CampaignAction</li> </ul> <p>Refer to <a href="#">Statistical Subjects</a> for a description of these values.</p> <p>Changes Take Effect: When Stat Server restarts.</p> <div> <p><b>Important</b></p> <p>The AgentStatus and PlaceStatus objects were synonymous in releases 5.1, 6.0, and 6.1. However, they are independent in 6.5 and later releases.</p> </div> |      |
| <business attribute> | <p>Specifies business attributes that Stat Server applies as a filter during its computation of statistics. The value string is recognized as a single value only. Starting with release 7.1, Stat Server supports the MediaType business attribute. Specifying this option is not mandatory.</p> <p>Default Value: No default value</p> <p>Valid Values: Non-empty string</p> <p>Changes Take Effect: When Stat Server restarts.</p>  | Yes  |

| Option | Description   | Java |
|--------|---|------|
|        | <p>The name of the business attribute must be a valid business attribute that is already defined to a particular tenant before Stat Server starts. This name cannot coincide with the reserved names for other Stat Server configuration options, such as <b>Subject</b>, <b>Category</b>, and <b>Filter</b>. Furthermore, the name must not contain special symbols (such as  , =, or ;) or spaces.</p> <p>Starting with the 8.5.105 release, Stat Server supports <b>multiple business attribute values</b>, specified as a comma-separated list for a given business attribute, in the stat type definition. Comma in the list of business attribute values is working as or in the related filter. Business attributes in different options of the Stat Type are working as different independent filters. This is applicable to all statistical categories, except Compound Categories and JavaCategory.</p> <p><b>Important</b></p> <ul style="list-style-type: none"><li>• There are no restrictions on using business attributes with stat types that have only instant and/or retrospective actions in MainMask and ReMask.</li><li>• Mutable business attributes (those that can be changed during the life of interaction), such as Customer Segment, should not be used with stat types that have durable actions in MainMask and/or ReMask due to unpredictable statistical results.</li><li>• Modification of business attributes (and/or their values) in the configuration, while Stat Server is running, might cause unpredictable statistical results.</li><li>• Business attributes usage when Subject is set to DNStatus or</li></ul> |      |

| Option                          | Description  | Java |
|---------------------------------|--|------|
|                                 | <p>PlaceStatus or AgentStatus is not recommended due to unpredictable statistical results.</p>   |      |
| ReasonStartOverridesStatusStart | <p>Determines how Stat Server computes current-state statistics. If this option is set to no, Stat Server uses the timestamp that is affiliated with the agent's current status, as in prior releases, to determine statistical values. If this option is set to yes, Stat Server also considers the timestamp that is affiliated with changes in reason code.</p> <p>Default Value: no</p> <p>Valid Values: yes, no</p> <p>Changes Take Effect: When Stat Server restarts.</p> <p>Setting this option to yes enables Stat Server to provide more refined results for those circumstances in which agents designate different reasons for being in the same state. This option is applicable only to the CurrentStateReasons statistical category. The Subject has to be set to DNAction for monitoring reasons changes.</p> |      |
| UseSourceTimeStamps             | <p>For those metrics that qualify, this option specifies whether Stat Server uses the actual time that events were transmitted to Stat Server (source timestamp) or the time that Stat Server acknowledges receipt of the events (the default behavior) when calculating metric duration. Setting this option to yes enables better consistency with the metrics provided by Interaction Concentrator (ICON) and other downstream Genesys applications of ICON.</p>  |      |



| Option  | Description   | Java |
|---------|---|------|
|         | <p>Qualifying metrics have both of the following characteristics:</p> <ul style="list-style-type: none"><li>• <code>[TimeProfileName]=Selection</code> or <code>Growing</code></li><li>• <code>[StatTypeDef]</code><br/><b>Subject</b>=<code>DNAction</code> or <code>CampaignAction</code><br/><b>MainMask</b>=one or more durable and/or retrospective actions (including instantaneous actions that carry an associated duration, like <code>AgentLogin</code>).<br/><b>Category</b>=historical categories only, with the exceptions of <code>JavaCategory</code> and <code>TotalAdjustedTime</code>.</li></ul> <p>Stat Server ignores a yes value for this option if the metric fails the qualification test.</p> <p>Default Value: no</p> <p>Valid Values: no, yes</p> <p>Changes Take Effect: When Stat Server restarts.</p> <p>Refer to <a href="#">Stat Server Timestamps</a> for an extended discussion of Stat Server's use of source timestamps.</p> |      |
| Formula | <p>Enables Stat Server to compute user-specific quantities that are based on attached data communicated by TEvents. The <a href="#">Custom Formulas</a> chapter is dedicated to an extended discussion of this subject. You can define a custom formula as described in the <a href="#">Custom Formulas</a> section below.</p> <p>A special specifier—<code>DistByConnID</code>—affects Stat Server's mechanism of aggregating statistics for the call-related actions that are listed in the main mask.</p> <p><code>DistByConnID</code> is applicable only to the limited number of statistical categories:</p>   |      |

| Option | Description   | Java |
|--------|---|------|
|        | <ul style="list-style-type: none"><li>• TotalNumber</li><li>• TotalAdjustedNumber</li><li>• CurrentNumber</li><li>• TotalTime</li></ul> <p>When the DistByConnID specifier is used in a stat type's definition, Stat Server groups the statistic's actions by connection ID (ConnID). In general, the contribution of a group of actions differs from that of the sum of contributions of the individual actions in that group—as is the case when DistByConnID is not specified for a statistic. <b>For example</b>, three agents are participating in the internal conference call. The ConnID is the same for all three agents. We want to calculate the value of the CurrentNumber statistic for the CallInternal action for the group of these agents. If DistByConnID is applied, the result is 1, otherwise the result is 3.</p> <p>Stat Server's procedure of grouping actions by connection ID applies to the actions specified in <b>MainMask</b> for the objects that are associated with the statistic. The procedure differs for each statistical category and is described as follows:</p> <ul style="list-style-type: none"><li>• For the TotalNumber/TotalAdjustedNumber statistical categories, when any of the following conditions are true, Stat Server increments the statistic at the end of an action or the start of status respectively:<ul style="list-style-type: none"><li>• An action or status with a particular ConnID starts.</li><li>• There are no actions or statuses in progress for the same ConnID.</li><li>• No such actions or statuses were in progress for less than one minute ago (1 minute is hard-coded).</li></ul></li></ul> |      |

| Option | Description  | Java |
|--------|--|------|
|        | <p>If the action or status is unrelated to a call, then aggregation functions in the same manner as when <code>DistByConnID</code> is not specified.</p> <div><b>Important</b><p><code>DistByConnID</code> is not applicable in <code>TotalNumber/TotalAdjustedNumber</code> statistics where filters or time ranges are also defined. Therefore, using <code>DistByConnID</code> with the <code>TotalNumber/TotalAdjustedNumber</code> statistical categories together with filters or time ranges may produce wrong results.</p></div> <ul style="list-style-type: none"><li>For the <code>CurrentNumber</code> statistical category, when either of the following conditions is true, Stat Server increments the statistic:<ul style="list-style-type: none"><li>An action or status with a particular <code>ConnID</code> starts.</li><li>There are no actions or statuses in progress for the same <code>ConnID</code>.</li></ul></li></ul> <p>When the action or status with the particular <code>ConnID</code> ends, Stat Server decrements the statistic only if there are no more actions or statuses in progress for that <code>ConnID</code>.</p> <p>If the action is either not call-related or not durable, Stat Server ignores this action in statistic calculations.</p> <ul style="list-style-type: none"><li>For the <code>TotalTime</code> statistical category, the group of actions or statuses in progress for a particular <code>ConnID</code> yields a one-second increment to the statistic for each second of</li></ul> |      |

| Option | Description   | Java |
|--------|---|------|
|        | <p>the group's existence. Where the statistic's <b>Subject</b> is other than DAction, Stat Server immediately reflects this increment in the statistical value. Where <b>Subject</b>=DAction, Stat Server updates the statistic's value in a stepwise fashion, incrementing the statistic when the oldest action belonging to a group of actions ends. If an action is either not call-related or not durable, Stat Server ignores this action in statistic calculations.</p> <p><b>Tip</b></p> <ul style="list-style-type: none"><li>• If you use the DistByConnID specifier, you must list it first among the <b>Formula</b> values as such:<br/>Formula=DistByConnID,...</li><li>• Stat Server recognizes the following aliases for DistByConnID:<ul style="list-style-type: none"><li>• DistinguishByConnID</li><li>• DCID</li></ul></li><li>• Any filtering that might be used in conjunction with a statistic, such as the designation of a MediaType, is applied <i>prior</i> to Stat Server's processing of DistByConnID.</li><li>• When used with DistByConnID, CurrentNumber and TotalTime categories ignore actions that are not associated with T-Server calls.</li></ul> |      |

| Option                     | Description   | Java |
|----------------------------|---|------|
|                            | <ul style="list-style-type: none"><li>The TotalNumber category, used with the DistByConnID specifier, considers actions that are not associated with T-Server calls as having a unique ConnectionID. Therefore, those actions are taken into account.</li></ul> <p>Default Value: No default value</p> <p>Valid Values: DistByConnID, DCID, DistinguishByConnID, <i>&lt;custom formula&gt;</i></p> <p>Changes Take Effect: When Stat Server restarts.</p> <p>Starting with release 8.5.102, the following features are available in formulas:</p> <ul style="list-style-type: none"><li>GetList function and IsNull predicate.</li><li>Keywords true and false.</li><li>Inline if: <code>&lt;boolean expression&gt; ? &lt;result1&gt; : &lt;result2&gt;</code>.<br/>If boolean expression is true then result1 is returned, else result2 is returned.</li><li>DNIS and ThisQueue can be compared to a variable string expression.</li></ul> |      |
| ApplyFilterAtActionEndOnly | <p>With this option set to yes, Stat Server checks the filter on a durable action at its end and if it is true, then the whole action duration (not a sub-period while the filter was true) is used in a statistical algorithm.</p> <p>This option is only taken into account for Subject=DNAAction or Subject=CampaignAction and only for Growing and</p>  |      |

| Option | Description   | Java |
|--------|---|------|
|        | <p>SlidingWindow aggregation intervals.</p> <p>Default Value: no</p> <p>Valid Values: yes, no</p> <p>Changes Take Effect: When Stat Server restarts.</p> <p><b>Important</b></p> <p>The ApplyFilterAtActionEndOnly specifier:</p> <ul style="list-style-type: none"><li>• Can be used with durable, instant, retrospective actions (for instant and retrospective actions, the statistic behavior with or without the specifier is identical).</li><li>• Is not taken into account for current statistical categories.</li><li>• Is not taken into account for the following (historical) statistical categories:<ul style="list-style-type: none"><li>• ServiceFactor1</li><li>• JavaCategory</li><li>• AverageOfCurrentNumber</li><li>• AverageOfCurrentTime</li><li>• EstimWaitingTime</li><li>• ExpectedWaitTime2</li><li>• LoadBalance</li><li>• MinNumber</li></ul></li></ul> |      |

| Option                | Description  | Java |
|-----------------------|--|------|
|                       | <ul style="list-style-type: none"><li>• MaxNumber</li><li>• TotalNumberErrors</li><li>• TotalAdjustedTime (if mode=ResetBasedNotification only)</li><li>• TotalCustomValue</li><li>• MinCustomValue</li><li>• MaxCustomValue</li><li>• AverageCustomValue</li></ul>  |      |
| Description           | <p>Specifies a description for this stat type. Specifying this option is discretionary; Stat Server ignores any value that you set for this option.</p> <p>Default Value: No default value</p> <p>Valid Values: String of fewer than 256 characters</p> <p>Changes Take Effect: When Stat Server restarts.</p> | Yes  |
| DynamicOverloadPolicy | <p>Defines actions that Stat Server may apply to a given statistic to reduce the <b>overload</b>.</p> <p>Default Value: 0</p> <p>Valid Values:</p> <ul style="list-style-type: none"><li>• 0 - sends and updates for requested statistics can be cut;</li></ul>  |      |

---

| Option           | Description  | Java |
|------------------|--|------|
|                  | <ul style="list-style-type: none"><li>• 1 - only sends of statistics to Stat Server clients can be cut;</li><li>• 2 - nothing can be cut, Stat Server updates and sends all requested statistics.</li></ul> <p>Changes Take Effect: When Stat Server restarts.</p> |      |
| <any other name> | Defines a custom parameter (specific option) for the stat-type with <b>Category</b> set to JavaCategory.   |      |



### Important

- If you want to change the definition of a stat type during runtime, you must first delete the entire stat-type definition and then re-create it with its new definition. Otherwise, Stat Server will recognize the change only upon restart.
- Stat Server clients may recognize other options for stat types that are not listed in the Table above. For instance, Data Sourcer requires that the **AggregationType** option be specified for statistics derived from a Stat Server Java extension. This information is processed by the client; Stat Server ignores such options.

## Classification of Statistical Types

Statistical types can be classified in distinct groups—for example:

- Status-based statistics.
- Interaction-related statistics.

Status-based statistics reflect changes in object statuses and generally contain the word *status* in their names. Interaction-related statistics reflect the telephony or multimedia information applied to specific objects, and characterize the interaction flow passing through the objects. Additional statistics, such as ExpectedWaitTime and LoadBalance statistics, reflect other characteristics of the contact center that are not related to status changes or telephony object information.

In addition, you can classify statistics based on any part of their stat type definition, such as their type of filter, object, and/or subject, or on any other criteria that you specify.

## Custom-Value Statistical Types

Custom-value stat types improve business data reporting by enabling you to define statistics that use formulas specific to your needs. Using your own formulas, you can create statistics that calculate average sales revenue per call and the total sales revenue for a specific time interval. The custom-value stat types that you define then become available to client applications that request them.

The format of custom-value stat types is similar to the format of Genesys-provided stat types. Custom-value stat types, however, lack the **RelMask** option and always contain the **Formula** option for which you must supply a value. At the [top](#) of this page see a description of the predefined statistical type format.

The Table below shows the statistical categories that apply to custom-value statistics.

**List of Custom-Value Categories**

| Historical | Current |
|------------|---------|
|------------|---------|

---

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• TotalCustomValue</li><li>• AverageCustomValue</li><li>• MinCustomValue</li><li>• MaxCustomValue</li></ul> | <ul style="list-style-type: none"><li>• CurrentCustomValue</li><li>• CurrentAverageCustomValue</li><li>• CurrentMinCustomValue</li><li>• CurrentMaxCustomValue</li></ul> |
|---|--|

These categories are described on the [Historical CustomValue Categories](#) and [Current CustomValue Categories](#) pages.

### Example

Suppose that you want to define a custom-value stat type that calculates the average sales revenue generated for every inbound call received by an agent. To accomplish this, create and define a new stat type section in the Stat Server Application object as follows:

1. Open the **Options** tab of the Stat Server application.
2. Create a new section and name it **AverSalesAmountPerInboundCall**, for example.
3. Within this section, add the **Objects** option and set its value to Agent, Place, GroupAgents, GroupPlaces.
4. Add the **Category** option to this section and set its value to AverageCustomValue.
5. Add the **MainMask** option and set its value to CallInbound.
6. Add the **Subject** option and set its value to DNAction.
7. Add the **Formula** option and set its value to `GetNumber("Price", 1) * GetSum("Amount")`. (Refer to [Custom Formula](#) below for an explanation of this formula.)
8. Apply the changes.

A configuration-file export of this section, as defined, appears as follows:

```
[AverSalesAmountPerInboundCall]
Objects=Agent, Place, GroupAgents, GroupPlaces
Category=AverageCustomValue
MainMask=CallInbound
Subject=DNAction
Formula=GetNumber("Price", 1) * GetSum("Amount")
```

### Custom Formulas

#### Important

For an evaluation of custom formulas, refer to [Custom Formulas](#).

**Note:** Custom formulas can be requested with **Subject=DNAction** only.

Custom formulas define custom values from an action on the basis of attached data. Attached data

can be attached to the call by different T-Server clients. An IVR might attach data to a call, for example, by collecting the numbers that callers press on their telephone keypads in response to a prompt. An agent might also attach data to a call using a desktop application. The language used in custom formulas is similar to that used in filters. Each formula is an arithmetic expression built from function calls and numeric constants, consisting of:

- Function calls. Custom formulas can use values from the key-value UserData lists received with TEvents related to Stat Server actions. Access to these values is provided by the functions listed in the **Key-Value List Functions in Custom Formulas** table below. Note that the list can include more than one pair with the same key.
- Operators, as well as parentheses (for suppressing standard precedence rules).

**Operators in Custom Formulas**

| Operator | Description            |
|----------|------------------------|
| +        | Addition               |
| -        | Subtraction            |
| /        | Division               |
| *        | Multiplication         |
| ?        | Then (in an Inline IF) |
| :        | Else (in an Inline IF) |

- Numeric constants.

Custom formulas always return a value of type float. The returned value is used in statistical calculations for each category.

### Important

You can apply filters to custom-formula statistics too.

The Table below lists functions to access key-value UserData lists. Local key-value lists function with data attached at the DN where the action occurs. Global key-value lists function with data attached at all participating DNs during the call.

### Important

For momentary actions, the GetGlobalMax function returns the same value as the GetMax function.

**Key-Value List Functions in Custom Formulas**

| Local Functions (Used for Local Key-Value List Calculations) |   |
|--|---|
| Function   | Description   |
| GetNumber("Key", Index)                                      | Returns the numeric value of the occurrence of the given key as specified by Index: |

| Local Functions (Used for Local Key-Value List Calculations)   |   |
|--|---|
|  | <ul style="list-style-type: none"> <li>• If Index is -1, the last occurrence is used.</li> <li>• If Index is a positive integer n, the <i>n</i>th occurrence is used.</li> </ul> <p>When Index exceeds the total number of occurrences of the given key in the list, or the key does not occur in the list at all, the returned value is 0 (zero).</p> <p>Index is an optional attribute for this property. If not specified, Stat Server substitutes -1 for its value; hence, <code>GetNumber("Key")</code> is equivalent to <code>GetNumber("Key", -1)</code>.</p>      |
| <code>GetMax("Key")</code>                                     | Returns the maximum value among all the values of pairs with the given key. When there are no such pairs, 0 is returned.  |
| <code>GetMin("Key")</code>                                     | Returns the minimum value among all the values of pairs with the given key. When there are no such pairs, 0 is returned.  |
| <code>GetSum("Key")</code>                                     | Returns the sum of all the values of pairs with the given key. When there are no such pairs, 0 is returned.   |
| <code>GetAver("Key")</code>                                    | Returns the average of all the values of pairs with the given key. When there are no such pairs, 0 is returned.   |
| <code>GetList( &lt;List&gt;, "Key", "Value" )</code>           | Returns a list of data and can be used in other functions that are able to process any list of data.<br><b>Example:</b> <code>MyFilter1=PairExists( GetList( UserData, "key1" ), "key2", "*" )</code> .   |
| <code>IsNull( &lt;Parameter&gt; )</code>                       | Returns true or false. The parameter can be one of the following: <code>UserData</code> , <code>ExtensionReasonCode (Reason)</code> , <code>Reasons</code> , or <code>Extensions</code> .   |
|  | <b>Note:</b> Starting with release 8.5.102, <code>GetAver</code> , <code>GetSum</code> , <code>GetMin</code> , <code>GetMax</code> , and <code>GetNumber</code> are applicable not only to <code>UserData</code> but to any key-value list attributes: <code>UserData</code> , <code>Reasons</code> , <code>ExtensionReasonCode (Reason)</code> , <code>Extensions</code> , <code>System</code> , or a list returned by the <code>GetList</code> function.  |
| Global Functions (Used for Global Key-Value List Calculations) |   |
| <code>GetGlobalNumber("Key", Index)</code>                     | <p>Returns the numeric value of the occurrence of the given key, attached at any DN, which is a member of the call, as specified by Index:</p> <ul style="list-style-type: none"> <li>• If Index is -1, the last occurrence is used.</li> <li>• If Index is a positive integer n, the <i>n</i>th occurrence is used.</li> </ul> <p>When Index exceeds the total number of occurrences of the given key in the list, or the key does not occur in the list at all, 0 is the returned value.</p> <p>Index is an optional attribute for this property. If not specified,</p> |

| Local Functions (Used for Local Key-Value List Calculations) |  |
|--|--|
|  | Stat Server substitutes -1 for its value; hence, <code>GetGlobalNumber("Key")</code> is equivalent to <code>GetGlobalNumber("Key", -1)</code> .                              |
| <code>GetGlobalMax("Key")</code>                             | Returns the maximum value among all the values of pairs, attached at any DN, which is a member of the call, with the given key. When there are no such pairs, 0 is returned. |
| <code>GetGlobalMin("Key")</code>                             | Returns the minimum value among all the values of pairs, attached at any DN, which is a member of the call, with the given key. When there are no such pairs, 0 is returned. |
| <code>GetGlobalSum("Key")</code>                             | Returns the sum of all the values of pairs, attached at any DN, which is a member of the call, with the given key. When there are no such pairs, 0 is returned.              |
| <code>GetGlobalAver("Key")</code>                            | Returns the average of all the values of pairs, attached at any DN, which is a member of the call, with the given key. When there are no such pairs, 0 is returned.          |

**Example 1:** Suppose that you want to multiply 99.99 by the sum of all the values of key-value pairs with key "Amount". To do so, enter the following formula:

```
99.99 * GetSum( "Amount" )
```

### Example 2:

To count interactions that have been placed by an agent in an Interaction Queue the following stat type can be used:

```
Category=TotalCustomValue
```

```
Objects=Agent
```

```
MainMask=CallInbound,CallOutbound,CallInternal
```

```
Formula=PairExists( System, "Operation", "place_in_queue" ) ? 1 : 0
```

```
Subject=DNAAction
```

### Important

`PairExists()` without a key-value list indication does not search through **System attributes**.

## GroupBy Feature

Starting with release 8.5.103, Stat Server supports the GroupBy stat type specifier. GroupBy defines

the mapping of an action to a group. Each action belongs to only one group at any moment of time.

GroupBy statement is the comma separated expression list. An expression can be of types:

- bool, for example the PairExists function.
- float and int, for example GetNumber, GetMin, GetMax, and other functions from the [Key-Value List Functions in Custom Formulas](#) table.
- string, for example the GetString function.

The result value for each group is calculated individually and is defined by Category. A special token null, which indicates an absence of a key, is a possible value for an expression.

There are three optional specifiers that can be used with GroupBy in stat types:

- **OrderBy**  
**OrderBy** is the comma separated list of tokens. Each token is a positive or negative integer, that refers to an index of the element in the GroupBy expression.

### Tip

Indexes start with 1.  
If OrderBy is not provided, the groups come unordered.  
Use the minus (-) sign to sort in descending order.

- **RowCount**  
**RowCount** is an integer stat type specifier that allows you to limit the number of returned groups. Its valid values are from 0 (zero) to 2147483647. If RowCount is not specified, the default value of 2147483647 is used.  
Note that, in some scenarios, Stat Server can ignore the RowCount set limit when loading saved aggregate data from backup files, resulting in additional groups being restored.
- **GroupByColumns**  
**GroupByColumns** specifies column names, corresponding to GroupBy. Stat Server sends this information to its clients in the GroupByColumns key in the embedded key-value list, associated with GroupBy StatType.

### Important

- Only the DNAction Subject is allowed for GroupBy.
- GroupBy is not compatible with DistByConnID in formulas.
- Legacy clients, such as CCPulse+, do not support GroupBy.
- Filters can be used in combination with GroupBy.
- GroupBy stat types are serialized into/from backup file, as well as other statistics.
- GroupBy is supported with old (dynamic) API as well.

The following categories cannot be used with GroupBy:

- CurrentState
- CurrentStateReasons
- CurrentTargetState
- JavaCategory
- ServiceFactor1
- TotalNumberErrors
- EstimWaitingTime
- LoadBalance
- EstimTimeToComplete
- Formula
- ExpectedWaitTime2
- CurrentDistinctNumber
- TotalDistinctTime

GroupBy cannot be used with the following statistics:

| Categories    | Object      | Subject        | Action              |
|---------------|-------------|----------------|---------------------|
| CurrentNumber | Campaign    | CampaignAction | RecordsNotProcessed |
| CurrentNumber | CallingList | CampaignAction | RecordsNotProcessed |
| CurrentNumber | Campaign    | CampaignAction | RecordsScheduled    |
| CurrentNumber | CallingList | CampaignAction | RecordsScheduled    |

If the queue-use-pseudo-actions configuration option is set to true, GroupBy cannot be used with the following statistics:

---

| Categories                      | Object                         | Subject  | Action                   |
|---------------------------------|--------------------------------|----------|--------------------------|
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | CallWait                 |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | AgentLogin               |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | AgentActive              |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | AgentReady               |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | DNLogin                  |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | DNActive                 |
| CurrentNumber                   | Queue, RoutePoint, GroupQueues | DNAction | DNReady                  |
| CurrentRelativeNumberPercentage | Queue, RoutePoint, GroupQueues | DNAction | any of the above actions |



**Example 1.**

If you want to know the total time each agent was active while logged into a queue, you can use the following stat type:

```
Category = TotalTime  
Objects = Queue, GroupQueues  
Subject = DNAction  
MainMask = AgentActive  
GroupBy = GetString(System, "AgentID")
```

For the above statistic you will get a result, similar to below:

|              |             |
|--------------|-------------|
| Total value: | 120 seconds |
| Agent_1      | 25 seconds  |
| Agent_2      | 50 seconds  |
| Agent_3      | 45 seconds  |

**Example 2.**

If you want to know how many calls came from each customer on some Agent Group, assuming that the customer name is attached as User Data in the customer\_name key, you can use the following stat type:

```
Category = TotalNumber  
Objects = GroupAgents  
Subject = DNAction  
MainMask = CallAnswered  
GroupBy = GetString(UserData, "customer_name")
```

For the above statistic you will get a result, similar to below:

|              |    |
|--------------|----|
| Total value: | 10 |
| Customer_1   | 3  |
| Customer_2   | 5  |
| Customer_3   | 2  |

**Example 3.**

If User Data contains some additional information about customers, like country and city in corresponding keys and you want to get calls distribution for these parameters, you can use the following stat type:

```
Category = TotalNumber  
Objects = GroupAgents  
Subject = DNAction  
MainMask = CallAnswered  
GroupBy = GetString(UserData, "customer_name"), GetString(UserData, "country"),  
GetNumber(UserData, "city")
```

For the above statistic you will get a result, similar to below:

|              |               |               |   |
|--------------|---------------|---------------|---|
| Total value: | 10            |               |   |
| Customer_1   | United States | New York      | 3 |
| Customer_2   | Canada        | Toronto       | 5 |
| Customer_3   | United States | San Francisco | 2 |

If you need to order resulted groups by country (the second item in a group) in ascending order add `OrderBy=+2` to the stat type definition.

If you need to see only first 2 groups, add `RowCount=2` to the stat type definition.

## DistinguishBy

Starting with release 8.5.104, Stat Server supports the `DistinguishBy` stat type specifier to prevent duplicate counts for the `CurrentNumber` statistical category, based on the action's content.

Stat Server applies an expression, defined by `DistinguishBy`, to any action in the `MainMask`. Therefore, Stat Server maps an action to a group.

### Important

- Only the `DNAAction Subject` is allowed for the `DistinguishBy` specifier.
- The `DistinguishBy` specifier is only supported for the `CurrentNumber` statistical category.
- `Formula=DCID` is ignored, if specified in addition to `DistinguishBy`.
- The `GroupBy` specifier is ignored, if specified in addition to `DistinguishBy`.
- The `DistinguishBy` specifier can also be used with the dynamic (old) API.

### Example 1.

The following statistic calculates the number of agents associated with a `GroupQueues` via the login into a mediation DN, which belongs to this `GroupQueues`. The agent, logged into multiple queues, is counted only once. Make sure, that the `[statserver]/queue-use-pseudo-actions` option is set to `false`.

```
Category=CurrentNumber
Objects=GroupQueues
MainMask=AgentLogin
Subject=DNAAction
DistinguishBy=GetString( System, "AgentID" )
```

### **Example 2.**

The following statistic calculates the current number of distinct pairs (<interaction type>, <key value>) extracted from Handling actions on an agent group.

```
Category=CurrentNumber  
Objects=GroupAgents  
MainMask=Handling  
Subject=DNAction  
DistinguishBy=GetString( System, "InteractionType" ), GetNumber( GlobalUserData,  
"key2" )
```