Stat Server User's Guide

Filters Section

12/23/2025

Filters Section

Contents

The [Filters] section of the Stat Server application defines conditions for excluding call- and non-call-related activity based on certain criteria specified in a logical condition. If used, this section must be named Filters. Filters allow you to restrict Stat Server actions taken into account during the computation of aggregate values. In a filtered statistic, Stat Server only considers those actions that satisfy a filter condition on certain attributes of TEvents, such as DNIS, ANI, CustomerID (or TenantID), MediaType, ThisQueue, TreatmentType, UserData, GlobalUserData, Extensions, Reasons, and ExtensionReasonCode. Stat Server also allows filtering by Interaction Server-driven events via the UserData and Reasons attributes.

Stat Server also considers the type of action in its analysis of a filter condition:

- For durable actions and statuses, Stat Server uses the number of times that a filter condition was true on an action (or status) and the duration of time for which the filter was true.

- For retrospective (instantaneous) actions, Stat Server evaluates a filter at the moment of action completion. If the filter condition is true, the statistic uses the entire duration of the action (and the number is 1).

This implementation does not change how Stat Server calculates Current statistics, but it does alter the calculation of historical statistics. Now, for example, instead of Stat Server returning the entire duration when an agent is NotReady with a particular reason only at the end of the NotReady state, Stat Server more accurately returns only that duration of time within the NotReady state for which the filter condition was true.
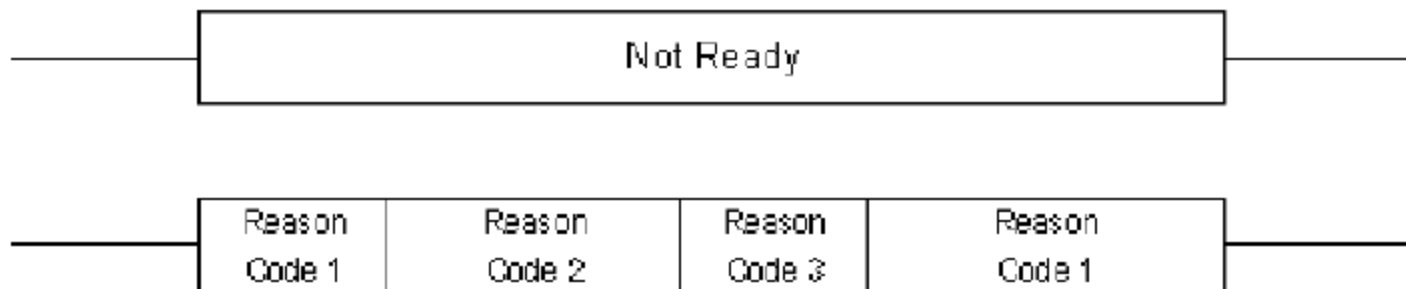
[+] Example

Assume that an agent has placed himself in the NotReady state for 50 minutes. During that state, he selected four reason codes for the following durations, respectively, on the phone set:

- ReasonCode 1—5 minutes

- ReasonCode 2—15 minutes

- ReasonCode 3—5 minutes

- ReasonCode 1—25 minutes

Using filter=ReasonCode 1, the current Stat Server implementation returns 2 as the number of times that filter=ReasonCode 1 or 30 minutes as the duration for which the filter condition was true during the NotReady state.

Previous implementations returned 1 as the number of times that the filter condition was true—only if filter = ReasonCode 1 was true at the moment that the agent left the NotReady state. Stat Server also returned 50 minutes, in this example, as the duration of time for which filter=ReasonCode 1.

Filter Example

The filters that you configure in Stat Server appear under the *Statistical Parameters* folder in Data Modeling Assistant (DMA), and among a particular statistic's properties within Pulse. (You can use DMA also to configure new filters.) If a Stat Server client requests a particular statistic with a filter, and that filter has been deleted from the configuration environment, Stat Server continues to calculate the statistic and sends the client an unfiltered value. Client applications can submit a statistic request that has no more than one filter applied.

Each opened statistic can have its own specific filter represented as a text string that contains a logical condition. The logical condition has to be proven for each call or device property. The result is either `true`, which includes the considered activity in the calculation, or `false`, which excludes the considered activity from the calculation. The logical condition has references to call or device properties, as well as to numeric and string constants, all of which are combined by operators.

Options in the `[Filters]` section consist of the following:

- *option name* – Any character string (no restrictions) that represents the name of the filter.

- *option value* – A logical condition that contains call or device properties and numeric or string constants that are combined by an operator. You can use the ? and/or * wildcard characters in the designation of the option's value. Stat Server matches ? in a wildcard string to any single character. Stat Server matches * to zero or more characters. The default value uses the `PairExists` function.

Important

- Wildcard characters in the designation of the option's value for the ExtensionReasonCode are supported only with the 3-argument filters. **Example:** `ExtRC1_a=PairExist(Extensions,"ReasonCode 1","*")`.

- For media channels, if there are multiple actions with the same priority, use filters with statistics where the Subject is `DNAction` or `Action`.
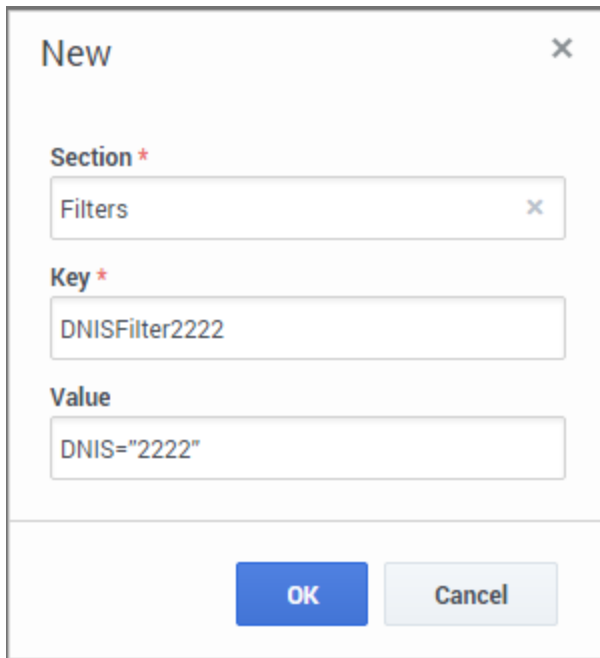
Configuration Option for Filters Section

Option	Description
<FilterName>	Defines a filter for filtering out call- and non-call-

Option	Description
	related activity, based on certain criteria that are specified in a logical condition. The logical expression is composed of: • Call or device properties • Operators • Values that consist of numerics, character string constants, or empty strings, depending on the call or device property. **Important** • The names of filters are unique only to the Stat Server application in which they are defined; they do not inherently reveal the tenant who created them. In multi-tenant environments that share the same Stat Server application, consider implementing a naming convention, such as TenantName-FilterName, to help users readily identify those filters that are pertinent to their branch of the business. • In addition, you must specify a value for this option; otherwise, Stat Server uses its default. Default Value: PairExists("*filtername*", "*") Valid Values: A logical expression Changes Take Effect: Immediately Stat Server recognizes the following functions as aliases for PairExists: • PairExist • TKVListPairExist • TKVListPairExists

[+] Example

Suppose that you want to set up a filter (DNISFilter2222) that considers calls whose Dialed Number Identification Service (DNIS) is 2222. To do so, in your Stat Server Application Options under the [Filters] section enter DNISFilter2222 in the Key field and DNIS="2222" in the Value field:

Defining a Filter

In this example, the call property is DNIS, the operator is the equal sign (=), and the constant is 2222.

Call Properties

Property Name	Operand Type	Description
DNIS	string	DNIS is the Dialed Number Identification Service. The DNIS is all or part of the telephone number that was dialed to make a call. Starting with release 8.5.102, DNIS can be compared to a variable string expression, in addition to a literal string.
ANI	string	ANI is the Automated Number Identification. The ANI is all or part of the caller's telephone number.
CustomerID	string	CustomerID is the tenant identification number as defined in the Configuration Layer.
MediaType	integer	MediaType identifies the media of interaction. For example, the media type of a call is voice. The predefined, case-sensitive media types are as follows:

Property Name	Operand Type	Description
		• 0 (voice)
		• 1 (voip)
		• 2 (email)
		• 3 (vmail)
		• 4 (smail)
		• 5 (chat)
		• 6 (video)
		• 7 (cobrowsing)
		• 8 (whiteboard)
		• 9 (appsharing)
		• 10 (webform)
		• 11 (workitem)
		• 12 (callback)
		• 13 (fax)
		• 14 (imchat)
		• 15 (busevent)
		• 16 (alert)
		• 17 (sms)
		• 100+ (custom)
		An elementary filter condition can contain either an integer value or a string with the predefined media type (for example, MediaType=5 or MediaType=chat)
ThisQueue	string	ThisQueue is the number of the queue. Starting with release 8.5.102, ThisQueue can be compared to a variable string expression, in addition to a literal string.
TreatmentType	string	The type of the treatment applied to a call, such as Silence, Music, Busy, and so forth. **Important** Use the Treatment key for this attribute in Stat Server filters. For example, Treatment=Busy.
UserData	string (TKVList)	UserData refers to the data that is attached to an interaction. An

Property Name	Operand Type	Description
		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. Or, an agent might attach data to a call from a desktop application. The `TKVList` refers to a set of functions that perform actions on `UserData` properties. K stands for *Key* and V stands for *Value*. T-Server sends attached data in key-value pairs; that is, one pair element specifies the key that describes the value, and the second element specifies the key's actual value. For example, `AfterCall` could be the name of a key, and the text `Processed the call for 10 minutes` could be the key's value. For memory, performance, and security reasons, Stat Server's processing of `UserData` strips the following types of `UserData` keys, which are not used for internal computations: • Keys included in at least one filter. • Keys coinciding with the names of business attributes. • Keys associated with the `EventUserEvent`, `EventPrivateInfo`, `EventError`, or `EventPartyInfo` TEvents. • GSW_RECORD_HANDLE, a predefined key used in the processing of user events for campaign-related statistic computations. **Important** Stat Server does not strip `UserData` containing the GSW_CALL_TYPE key because Stat Server uses this information to generate the ASM_Engaged action for agent DNs involved in outbound predictive dialing interactions. `UserData` that Stat Server uses for internal processing is packed into the values of `CurrentState` statistics.
GlobalUserData	string (TKVList)	`GlobalUserData` contains all user-data, associated with a given action, unlike `UserData`,

Property Name	Operand Type	Description
		which only contains the user-data, attached by the object, on which the action is generated.
Extensions	string (TKVList)	This property enables Stat Server to filter switch-specific and other features on any specified key-value pair recorded in the `Attribute Extensions` attribute of select TEvents. A filter using this property must be specified in the following format: `PairExists( Extensions, <key>, <value> )` where: `Extensions` is the hard-coded name of the `TKVList` function. *<key>* is a string representing the key of a key-value pair. *<value>* is an integer or string representing the *<key>*'s values. For example: `PairExists(Extensions,"Sales",10000)` (if the value is numeric) `PairExists(Extensions,"Color","Green")` (if the value is string) Stat Server applies a filter having this definition to a statistic for the following noncall-related TEvents that Stat Server receives from an agent's DN: • `EventAgentLogin` • `EventAgentLogout` • `EventAgentReady` • `EventAgentNotReady` • `EventDNDOn` • `EventDNDOff` • `EventRegistered` • `EventAddressInfo` Stat Server also applies a filter with this definition to the following call-related TEvents that Stat Server receives from regular DNs: • `EventAbandoned` • `EventAttachedDataChanged` • `EventDialing` • `EventEstablished`

Property Name	Operand Type	Description
		• EventHeld • EventNetworkCallStatus • EventPartyAdded • EventPartyChanged • EventPartyDeleted • EventPartyInfo (handled as EventEstablished) • EventQueued (handled as EventRinging) • EventReleased • EventRetrieved • EventRinging For call-related TEvents, filters using this property apply toward any associated actions. For noncall-related TEvents, only the following actions can be impacted by Extensions filtering: • AfterCallWork • LoggedIn • NotReadyForNextCall • WaitForNextCall

Device Properties

Property Name	Operand Type	Description
Reasons	string (TKVList)	Refers to additional data that is included in the TEvent to provide reasons for and results of actions taken by an agent. These reasons can originate from software- or hardware-related reasons—Stat Server does not differentiate between the two. Stat Server uses the value of the Reasons attribute in combination with the values of the UserData attribute when processing filters: PairExists(UserData, key, value) and PairExists(key, value)

Property Name	Operand Type	Description
		Stat Server uses the value of the Reasons attribute only in filters: `PairExists( Reasons, key, value )` When specified as such, Stat Server ignores any attached data that has `UserData` defined as the key in order to avoid consuming additional memory for its storage. **Important** Do not confuse this Reasons property with Reason, which serves as an alias for the `ExtensionReasonCode` property described in the next row.
ExtensionReasonCode	string	Refers to the reason code that T-Server propagates in its `AttributeExtensions` attribute of a TEvent. T-Server uses this key-value pair to gather switch-specific hardware reason codes that mostly accompany Ready and NotReady TEvents. Despite the fact that Stat Server does not restrict use of such variables in filters, Genesys recommends that you use filters with this variable only for accessing switch-related reason codes in non-call-related agent or DN states. Values of hardware reasons are switch-specific and must be configured on the customer side. In the event that T-Server propagates no reason code, Stat Server reports the value of this condition as Unknown and any filters using this property evaluate as False. Stat Server packs Reason attached data into the values of `CurrentState` statistics. Stat Server recognizes Reason as an alias of `ExtensionReasonCode`. This should not be confused with the Reasons property described in the row above. For example: • `ExtensionReasonCode = "Lunch"` (or `Reason = "Lunch"`) returns a True value if the value of the key-value pair returned by the ReasonCode key is equal to Lunch.

Property Name	Operand Type	Description
		• `ExtensionReasonCode != 12` (or `Reason != 12`) returns True if the `Extension` TEvent returns a key-value pair of ReasonCode (the key) and its accompanying value which is equal to a value other than 12. From 8.0 release, Stat Server supports less than or equal, greater than or equal, greater than, less than expressions for numeric operands. For example: • `ExtensionReasonCode >= 12` (or `Reason >= 12`) • `ExtensionReasonCode < 12` (or `Reason < 12`) **Important** Software reasons (propagated by the Reasons attribute) are still provided using the `PairExists` function. Stat Server does not differentiate between hardware- and software-related reasons.

Operators in Filters

Operators	Description
=	Equal (for strings or numeric operands)
!=	Not equal (for strings or numeric operands)
>=	Greater than or equal to (for numeric operands only)
<=	Less than or equal to (for numeric operands only)
>	Greater than (for numeric operands only)
<	Less than (for numeric operands only)
&	Logical AND
|	Logical OR
~	Logical NOT
( )	Parentheses (for changing operators' priorities)

Filter Expression Evaluations

The results of a filter expression can be TRUE, FALSE, or NULL; however, Stat Server returns to its clients either TRUE or FALSE depending on the expression's construction.

Filter sub-expressions, such as GetNumber(), may be evaluated to NULL if, for example, the referenced key in the key-value list does not exist—Stat Server cannot retrieve its value. NULL can also appear as a result of propagation. When evaluating filter expressions, Stat Server propagates NULL according to the following rules:

- Any arithmetical sub-expression having NULL as one of the operands, is evaluated to NULL (for example, NULL+2 yields NULL).

- Any comparison sub-expression having NULL as one of the operands, is evaluated to NULL (NULL=2 yields NULL).

In logical sub-expressions:

- NULL | TRUE yields TRUE.

- NULL & FALSE yields FALSE.

If the whole filter expression is evaluated to NULL, Stat Server returns FALSE as the final result.

Starting with release 8.5.102, the following features are available:

- GetList function and IsNull predicate in filters and formulas.

- Keywords true and false.

- Inline if: <boolean expression> ? <result1> : <result2>.
  If boolean expression is true then result1 is returned, else result2 is returned.
  Example:
  ~IsNull(Reason) ? (Reason>5) : false

- DNIS and ThisQueue can be compared to a variable string expression.
  Example:
  DNIS=GetGlobalNumber("DNNumber")
  ThisQueue != 8001

UserData

The key-value list UserData cannot be an operand of any operator. Instead, it can be listed as the first parameter of any one of the TKVList family of functions shown in the UserData Properties table, or it can be left out. For example,

PairExists(UserData,"key","value") and PairExists("key","value") are equivalent.

These filter function names can be preceded with TKVList, as was the case in previous versions of Stat Server. TKVListPairExists and PairExists are both valid names, for example.

Use the wildcard character * (asterisk) in place of the value in filter functions.
PairExists("Key","*") would return 1 for true if any key-value pair exists where the key equals

"Key", regardless of the value of that pair.

UserData Properties

Operators	Description
PairExists("Key", "Value")	Performs search for the specified pair. Returns a number: 1 (true) or 0 (false).
GetNumber("Key", Index)	Returns the numeric value of the occurrence of the given key as specified by Index: • If Index is -1, the last occurrence is used. • If Index is a positive integer n, the nth occurrence is used. 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 NULL. Index is an optional attribute for this property. If not specified, Stat Server substitutes -1 for its value; hence, GetNumber("Key") is equivalent to GetNumber("Key", -1).
GetString("Key", Index)	Returns the string of the value of the given key as specified by Index: • If Index is -1, the string of the last value is used. • If Index is a positive integer n, the string of the nth value is used. 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 NULL. Index is an optional attribute for this property. If not specified, Stat Server substitutes -1 for its value; hence, GetString("Key") is equivalent to GetString("Key", -1).
GetMax("Key")	Returns the maximum value among all pairs with this "Key". A value of NULL means that there are no pairs with the "Key".
GetMin("Key")	Returns the minimum value among all pairs with this "Key". A value of NULL means that there are no pairs with the "Key".
GetSum("Key")	Returns the sum of all values with this "Key". A value of 0 means that there are no pairs with the "Key".
GetAver("Key")	Returns the average of all values with this "Key". A value of 0 means that there are no pairs with the "Key".
GetList(<List>, "Key", "Value")	Returns a list of data and can be used in other functions that are able to process any list of data. **Example:** MyFilter1=PairExists(GetList(

Operators	Description
	UserData, "key1"), "key2", "*").
IsNull(<Parameter>)	Returns `true` or `false`. The parameter can be one of the following: UserData, ExtensionReasonCode (Reason), Reasons, or Extensions.

Note: Starting with release 8.5.102, GetAver, GetSum, GetMin, GetMax, GetNumber, and GetString are applicable not only to UserData but to any key-value list attributes: UserData, Reasons, ExtensionReasonCode (Reason), Extensions, System, or a list returned by the GetList function.

Examples:
```
GetAver( Reasons, "key" )
GetAver( Extensions, "key" )
GetAver( System, "key" )
```

For all functions dealing with numbers, the value of the key-value pair is evaluated as either an integer or a floating point. If the key type is an integer, the value is evaluated as an integer with no modifications. If the key type is a string, the value is a floating point. Constants for a logical condition can be either strings in double quotation marks ("English","3333") or numbers (100, 3.14). Numbers (constants and function return values) are floating-point values.

Starting with release 7.0, Stat Server ignores any attached data if no corresponding filter or custom-value formula has been defined within Stat Server that uses the specific key. This is done for performance and security reasons. Stat Server, furthermore, does not output attached data to the Stat Server log under this circumstance.

[+] Example

Suppose that you want to filter calls based on language. If the enterprise set up the key "Language" to identify language and the value "Spanish" for callers who speak Spanish, you could use the PairExists UserData function to search for calls with attached data in the key-value pair form of Language/Spanish.

On the Options tab of the Stat Server Properties dialog box, you could add a SpanishLanguage option in the [Filters] section and specify filtering for calls with attached data containing the key "Language" and the value "Spanish". The example would have SpanishLanguage in the Name field and PairExists("Language","Spanish") in the Value field.

Now, when an agent attaches the "Spanish/Language" key-value pair to calls from a desktop application, the calls are filtered out of statistical calculcations.

System Key-Value List

Stat Server supports System key-value list in filters.

The supported key-value list contains the following system attributes:

System Attribute Name	Value	Notes
AgentID	string	Supported on the AgentLogin,

System Attribute Name	Value	Notes
		AgentReady, and AgentActive mediation DN actions when the queue-use-pseudo-actions configuration option is set to `false`.
ActorType	strategy, agent, media_server	Supported on multimedia actions. The value is taken from the `attr_actor_type` event attribute. There are no benefits of using the `ActorType` attribute with durable actions.
budget_avail	integer	Available for Agents, Places, AgentGroups, and PlaceGroups on Routable and NotRoutable actions. Introduced in Stat Server release 8.5.110.03 to support budget-based routing.
budget_timestamp	integer	Available for Agents, Places, AgentGroups, and PlaceGroups on Routable and NotRoutable actions. Introduced in Stat Server release 8.5.110.20 to support budget-based routing.
budget_total	integer	Available for Agents, Places, AgentGroups , and PlaceGroups on Routable and NotRoutable actions. Introduced in Stat Server release 8.5.110.03 to support budget-based routing.
budget_used	integer	Available for Agents, Places, AgentGroups, and PlaceGroups on Routable and NotRoutable actions. Introduced in Stat Server release 8.5.110.03 to support budget-based routing.
ConnectionID	string	Applicable only for voice.
current_number	integer	Current number of interactions of given a media type on an agent/place. Applicable only to the Routable and NotRoutable actions.
InitialOperation	unknown, transfer, conference, intrude, route, pull, create, reject, timeout, leave, stop, place_in_workbin, place_in_queue, party_disconnect	Can be used in both `Filters` and `UserData` formulas. Applicable only for multimedia.
InteractionCost	integer	Available for call actions on DN, `Media Channel`, Queue, and `Routing Point`. Introduced in Stat Server release 8.5.110.03 to support budget-based routing.

System Attribute Name	Value	Notes
		The value of the `InteractionCost` attribute can change.
OrigInteractionCost	integer	Available for call actions on DN, `Media Channel`, Queue, and `Routing Point`. Introduced in Stat Server release 8.5.110.20 to support budget-based routing. The `OrigInteractionCost` immutable system attribute coincides with the value of the `InteractionCost` when the corresponding action is started.
InteractionID	string	Applicable only for multimedia.
InteractionSubtype	InboundNew, OutboundNew, etc.	Applicable only for multimedia.
InteractionType	Unknown, Inbound, Outbound, Consult, Internal	For multimedia, Unknown value is not applicable.
IsAccepted	yes, no	Applicable only for multimedia. Can have the yes value only in case of the first acceptance of an inbound interaction (Interaction Server event attribute `attr_itx_delivered_at` is equal to NULL).
IsOnline	yes, no	Applicable only for multimedia. Starting with release 8.5.104, the `IsOnline` system attribute is **no longer supported**.
max_number	integer	Maximum allowed number (according to a capacity rule) of interactions of a given media type on an agent/place. Applicable only to the Routable and NotRoutable actions.
media_state	1, 0	1 if media is ready and 0 otherwise. Applicable only to the Routable and NotRoutable actions.
MediaType	chat, email, voice, etc.	
Operation	unknown, transfer, conference, intrude, route, pull, create, reject, timeout, leave, stop, place_in_workbin, place_in_queue, party_disconnect	Should only be used in `UserData` formulas. Applicable only for multimedia.
RequestEnvelope	string	Associated with the `attr_esp_request_envelope` event attribute. Applicable only to the ExternalServiceRequested and ExternalServiceResponded actions.

System Attribute Name	Value	Notes
routable	integer	Number of interactions of a given media type that can be routed to an agent/place, according to a capacity rule. Applicable only to the Routable and NotRoutable actions. **Warning** Number of routable interactions for a given media is updated only by an interaction of this given media type.
ServiceObjective CompleteServiceObjective	duration in seconds	
VisibilityMode	unknown, conference, monitor, coach	Applicable only for multimedia.
WorkbinID	string	Name of the agent workbin into which the interaction is placed. It is defined by the `attr_itx_workbin_type_id` attribute from the Interaction Server event.
WorkbinOwnerID	string	Employee ID of the agent who owns the agent workbin into which the interaction is placed. It is defined by the `attr_itx_agent_id` attribute from the Interaction Server event.

Important

- All system attributes, with the exception of the `AgentID` attribute, in the Table above are supported starting with release 8.5.102. The `AgentID` attribute is supported starting with release 8.5.103. The `ActorType`, `current_number`, `max_number`, `media_state`, `RequestEnvelope`, and `routable` attributes are supported starting with release 8.5.104.

- Starting with release 8.5.104, the `IsOnline` system attribute is no longer supported.

- System attribute names are case-sensitive.

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

Example 1:
To calculate interactions with `InteractionType = Inbound` the following filter can be used:
`F1 = PairExists( System, "InteractionType", "Inbound" )`

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=DNAction
```

Example 3:
To count the number of chats, routable to an agent group the following stat type can be used:

```
Category=CurrentCustomValue
Objects=GroupAgents
MainMask=Routable
Subject=DNAction
Formula=GetNumber( System, "routable" )
```