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

Action Descriptions

5/8/2025

Action Descriptions

This page provides a list and descriptions of all the Actions in alphabetical order. If the same action can be generated on different objects, the action name includes the applicable object type in brackets. For lists of the Actions grouped by DN and Action types, see Regular DN Actions, Mediation DN Actions, Media-Channel Actions, StagingArea Actions, and Tenant Actions.

Accepted
Active
ACWCompleted
ACWMissed
AfterCallWork
AgentActive
AgentLogin (Regular DNs)
AgentLogin (Mediation DNs)
AgentLogout
AgentReady
ASM_Engaged
ASM_Outbound
Available
BeingCoached
BeingMonitored
Blocked
CallAbandoned
CallAbandonedFromDialing
CallAbandonedFromHold
CallAbandonedFromRinging (Regular DNs)
CallAbandonedFromRinging (Mediation DNs)
CallAbandonedFromRinging (Virtual Queues)
CallAnswered (Regular DNs)
CallAnswered (Mediation DNs)
CallAnswered (Virtual Queues)
CallCleared
CallConferenceJoined
CallConferenceMade
CallConferenceOriginated
CallConferencePartyAdded
CallConferencePartyDeleted
CallConsult
CallConsultCompleted
CallConsultOriginated
CallConsultReceived
CallConsultStarted
CallDialConferenced
CallDialTransferred
CallDialed
CallDialing
CallDialingStarted
CallDistributed
CallDistributedToQueue
CallEntered
CallForwarded (Regular DNs)
CallForwarded (Mediation DNs)
CallOutbound
CallOutboundCompleted
CallOutboundOriginated
CallOutboundReceived
CallOutboundStarted
CallPartyChanged
CallReleased (Mediation DNs)
CallReleased (Virtual Queues)
CallRetrievedFromHold
CallRinging
CallRingingPartyChanged (Regular DNs)
CallRingingPartyChanged (Mediation DNs)
CallRingingStarted
CallTransferMade
CallTransferPartyChanged
CallTransferTaken
CallTreatmentCompleted
CallTreatmentNotStarted
CallTreatmentStarted
CallUnknown
CallUnknownCompleted
CallUnknownStarted
CallWait
CoachingByIntrusionInitiated
CoachingByRequestInitiated
CoachingRequested
ConferenceJoined
ConferenceJoinedByIntrusion
ConferenceMade
DND
Delivering
DeliveringStarted
DNActive
DNLogin
DNReady
DoNotDisturb
ExternalServiceRequested (Tenants)
ExternalServiceResponded (Tenants)
Handling
HandlingInbound
HandlingInboundStarted
HandlingInternal
HandlingInternalStarted
HandlingOutbound
HandlingOutboundStarted
HandlingStarted
InteractionAnswered (Agent Workbin)
InteractionCleared
InteractionCleared (Agent Workbin)
InteractionCreated (StagingAreas)
InteractionCreated (Tenants)
InteractionDeleted (StagingAreas)
InteractionDeleted (Tenants)
InteractionDistributed
InteractionDistributed (Agent Workbin)
InteractionDistributedToQueue
InteractionEntered
InteractionEntered (Agent Workbin)
InteractionPastAcceptServiceLevel (StagingAreas)
InteractionPastAcceptServiceLevel (Tenants)
InteractionPastCompletionServiceLevel (StagingAreas)
InteractionPastCompletionServiceLevel (Tenants)
InteractionReleased (StagingAreas)
InteractionReleased (Tenants)
InteractionReleased (Agent Workbin)
InteractionResponded (Regular DNs)
InteractionResponded (MediaChannels)
InteractionWait (StagingAreas)
InteractionWait (Tenants)
InteractionWait (Agent Workbin)
LoggedIn
LoggedOut
Monitored (Regular DNs)
Monitored (Mediation DNs)
MonitoringInitiated
NotMonitored (Regular DNs)
NotMonitored (Mediation DNs)
NotReadyForNextCall
NotRoutable
OffHook
OnHook
OrigDNCallAbandoned
OrigDNCallDistributed
OrigDNCallEntered
OrigDNCallWait
Pulled
Rejected
Revoked
Routable

CallHeld
CallInbound
CallInboundCompleted
CallInboundStarted
CallInternal
CallInternalCompleted
CallInternalOriginated
CallInternalReceived
CallInternalStarted
CallMissed
CallObserved...
CallObserving...
CallOnHold

InteractionAbandoned (Tenants)
InteractionAbandonedDuringOffering (MediaChannels)
InteractionAbandonedDuringOffering (StagingAreas)
InteractionAbandonedDuringOffering (Tenants)
InteractionAbandonedDuringOffering (Virtual Queues)
InteractionAbandonedDuringOffering (Agent Workbin)
InteractionAccepted1
InteractionAccepted (StagingAreas)
InteractionAccepted (Tenants)
InteractionAccepted (Agent Workbin)
InteractionAgentPartyInProgress (Tenants)
InteractionAnswered (StagingAreas)
InteractionAnswered (Tenants)

StartedInternal
StartedOutbound
StoppedInbound
StoppedInternal
StoppedOutbound
StuckCallCleaned
StuckCallCleanedWhileRinging (Regular DNs)
StuckCallCleanedWhileRinging (Mediation DNs)
TransferMade
TransferTaken
TransferredFromHold
UserEvent (Regular DNs)
UserEvent (Mediation DNs)
UserEventReceived (Regular DNs)
UserEventReceived (Mediation DNs)
WaitForNextCall

Accepted

This retrospective action, also called `InteractionAccepted`, indicates that an agent (or place) has accepted a delivered interaction. This action terminates the Delivering action, and it is similar to CallAnswered in the telephony model.

Back to top

Active

This durable action tracks how long a media channel has been active for a particular agent (or place). Stat Server generates this action when the `EventMediaAdded` event is received from Interaction Server for the media on a place where an agent is logged in. This action ends with the `EventMediaRemoved` event from Interaction Server.

Back to top

ACWCompleted

This retrospective action occurs on a mediation DN when the regular DN action AfterCallWork is over. Action duration is the same duration as the corresponding `AfterCallWork` action. If a switch permits agents to enter `AfterCallWork` mode while they are still involved in calls, Stat Server generates the `ACW` on a regular DN upon completion of the interaction. Then, after the `ACW` action is ended, the `ACWCompleted` action is generated on a mediation DN, which distributes the interaction to regular DN. This behavior was introduced in the 7.0 release.

Stat Server generates an `ACWCompleted` or `ACWMissed` action on the mediation DN when the interaction is directed to the Position or Extension DN via a queue or routing point. This action was introduced in release 7.0.

Back to top

ACWMissed

This retrospective action occurs on a mediation DN when the regular DN action AfterCallWork is over. Action `ACWMissed` is generated on a mediation DN only if an agent enters ACW mode while s/he is on a call that was distributed from a source other than the mediation DN, on which the agent is logged in. Action duration is the same duration as the corresponding action `AfterCallWork`.

Back to top

AfterCallWork

This durable action is specific to particular switches and T-Server or SIP Server applications. For multimedia DNs, this action is classified as media-dependent, media-unique. While an agent is not involved in calls, this action starts when Stat Server receives `EventAgentNotReady` with a `WorkMode` attribute of `AfterCallWork` on any of the enabled media channels of a DN. Stat Server cancels generation of an `AfterCallWork` action (where it was previously postponed) if any of the following occur:

- Stat Server receives the EventAgentNotReady TEvent with a work mode other than AfterCallWork.
- Stat Server receives the EventAgentReady or EventDNDOn TEvents.
- Stat Server receives the EventAgentLogout TEvent (the agent logs out).

If a switch permits an agent to enter AfterCallWork mode while still involved in calls, any call ending with this agent will invoke after-call work. Stat Server generates the AfterCallWork action upon completion of the interaction. This behavior occurs even if Stat Server receives EventNotReady TEvent with Workmode=ACW from T-Server. Stat Server postpones the AfterCallWork action upon termination of the interaction.

The UserData, Reasons, and Extensions attributes from the EventDNDOn or EventDNDOff TEvents are not inherited by this action.

While AfterCallWork persists on a media channel of a multimedia DN, no routing is possible to that channel. (Stat Server marks the media_state component of the DN's capacity vector NR [NotReady].) Stat Server considers the actions occurring on all media channels when determining the DN’s status. A DN's status is the highest ranking action occurring on all enabled media channels according to Stat Server's status priority tables.

If Stat Server receives EventNotReady TEvent with Workmode=ACW while the interaction is active, this action is simultaneous with one of the following call-type actions:

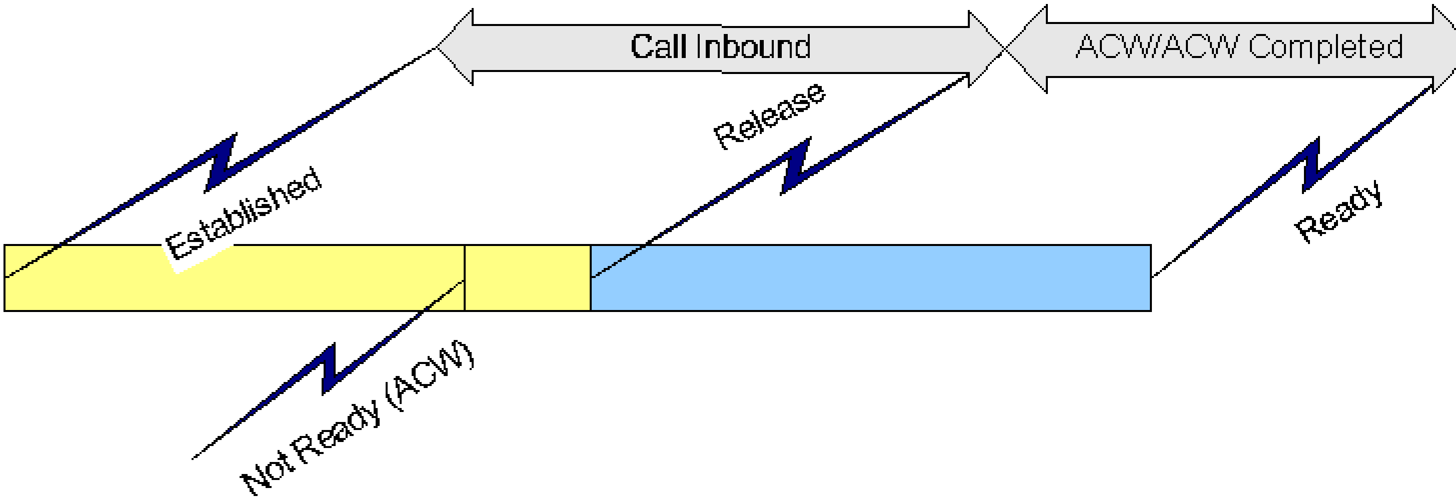
- AfterCallWorkUnknown
- AfterCallWorkInternal
- AfterCallWorkInbound
- AfterCallWorkOutbound
- AfterCallWorkConsult

The interaction type that Stat Server receives from T-Server together with EventReleased, determines which of the preceding five actions occurs simultaneously with AfterCallWork. If AfterCallWork starts after an interaction is released, none of these call-type actions occurs.

Starting with Release 7.0, Stat Server generates AfterCallWork actions only upon completion of the interaction. This behavior allows several statistics to be more independent from a T-Server (and/or a Desktop) implementation; one such statistic is that requested with the ACWCompleted action for queues.

> **Important**
>
> These changes do not affect Stat Server's MLink ACW emulation functionality, which is based on an entirely different TEvent set.

The diagram below illustrates the changes in the ACW calculation schema.



ACW Action Generated After Interaction Completion

> Tip
>
> See also DN Actions at Newly Registered DNs.

After-Call Work on the Nortel Meridian T-Servers

Stat Server processes ACW-related events when operating with the following T-Servers, subsequently referred to as Meridian or Meridian-like T-Servers:

- NEC-2400
- Release 7.0$^+$ of T-Server for Nortel Meridian 1
- Release 7.0$^+$ of T-Server for Nortel Symposium Call Center
- Release 7.1$^+$ of T-Server for Nortel Communication Server 1000 with SCCS/MLS

> Important
>
> The switch type you set in the Configuration Layer when working with these T-Server applications depends on your PBX type and can be `Nortel Meridian 1`, `Nortel Meridian CallCenter/Symposium`, or `Nortel Communication Server 1000 with SCCS/MLS`. Starting with Stat Server release 8.5.000.32, the configurable association between position and extension on the switch level is supported for many switches (for more information refer to the position-extension-linked option, which is configurable on the switch Annex). In this case, Stat Server considers related T-Servers as Meridian-like T-Servers.

Starting with release 7.0 of Meridian-like T-Servers, their ACW-related events are processed differently than they are with other T-Server types. The reason for the difference in processing is that the Meridian-like DN model is different from other DN models that Genesys supports. Unlike other models, this model consists of a Position and Extension DNs linked together.

- To indicate ACW, Meridian-like T-Server applications propagate an `EventAgentNotReady` TEvent with workmode=`ACW` the moment an agent requests after-call work functionality (that is, when he or she presses the ACW button). (Other T-Server applications propagate this TEvent upon completion or redirection of the interaction). Meridian-like T-Servers send this TEvent only for Position DN types—it does not send the event for Extension DNs. If no more than one Position/Extension pair is configured on a place, Stat Server logic links together Position and Extension DNs based on how the corresponding Place object is configured in Configuration Server.

- Meridian-like T-Servers propagate an additional `EventAgentNotReady` TEvent (workmode=ACW) if the agent changes the reason for being in ACW state.

- After-call work terminates when Stat Server receives from T-Server one of the following TEvents:
  - `EventAgentReady`
  - `EventAgentNotReady (workmode!= ACW)`
  - `EventAgentLogout`

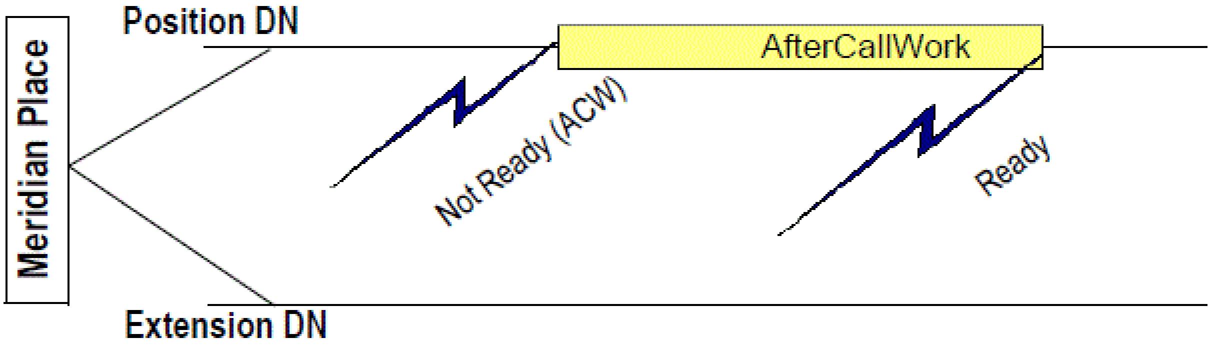
Based on the `EventAgentNotReady` TEvent (with workmode=ACW), Stat Server generates an `AfterCallWork` action on the Position DN and links the action with the appropriate telephony interaction, if applicable. In addition, if Stat Server recognizes this after-call work as associated with a particular telephony interaction, Stat Server postpones generation of the `AfterCallWork` action until the interaction is released. Furthermore, Stat Server inherits UserData keys and their values from the interaction and allows filtering of `AfterCallWork` action through these keys. If reasons are attached to `EventAgentNotReady` TEvent (workmode= ACW), then Stat Server can use them in filtering. Furthermore, Stat Server reacts when reasons change, such as upon receipt of the subsequent `EventAgentNotReady` ACW TEvent.

Stat Server generates an `ACWCompleted` or `ACWMissed` action on the mediation DN when the interaction is directed to the Position or Extension DN via a queue or routing point.

The following examples illustrate the actions Stat Server generates following receipt of certain TEvents from a Meridian-like T-Server.

ACW with No Associated Interaction

The diagram below illustrates a scenario where Stat Server immediately starts an `AfterCallWork` action on the Position DN upon receipt of the `EventAgentNotReady` TEvent (with workmode=ACW) from Meridian-like T-Server, and when there are no telephony interactions on the Position (or Extension) DN.



ACW Given No Telephony Interaction

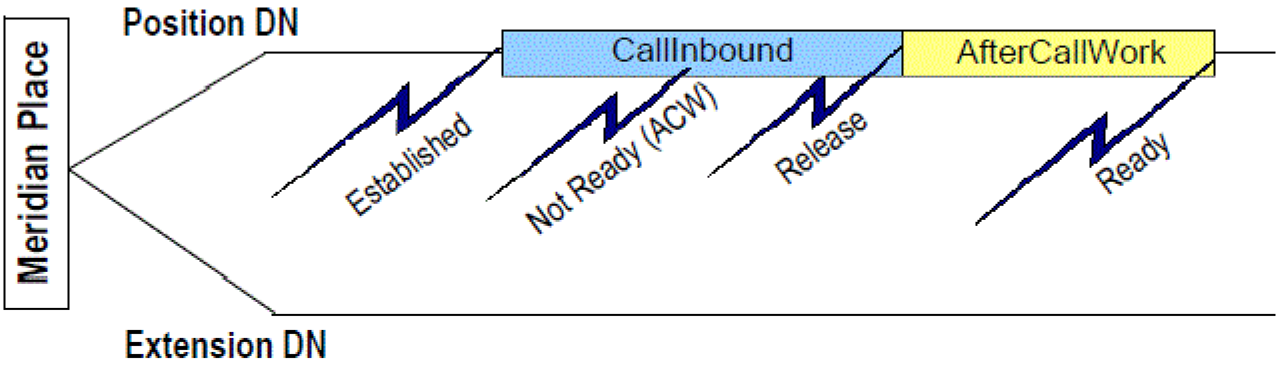
The diagram shows the events occurring on the Position DN where Stat Server starts an `AfterCallWork` action. Stat Server terminates it, in this example, upon receipt of an `EventAgentReady` TEvent. (The `EventAgentLogout` or `EventAgentNotReady` TEvents with workmode!= `ACW` would also terminate the action.)

> **Important**
>
> This scenario also applies for other T-Server applications that have only Position or only Extension DNs.

ACW Request During an Interaction

If a telephony interaction is currently in progress on a Position DN, the related Extension DN, or both, when Stat Server receives an ACW-related TEvent on that Position DN, Stat Server generates an `AfterCallWork` action on the Position DN only, and only after all calls complete on the Position and/or Extension DNs. Furthermore, Stat Server associates this action only with the last released interaction. Stat Server does not generate an `AfterCallWork` action on the Extension DN, regardless of where the last interaction took place.

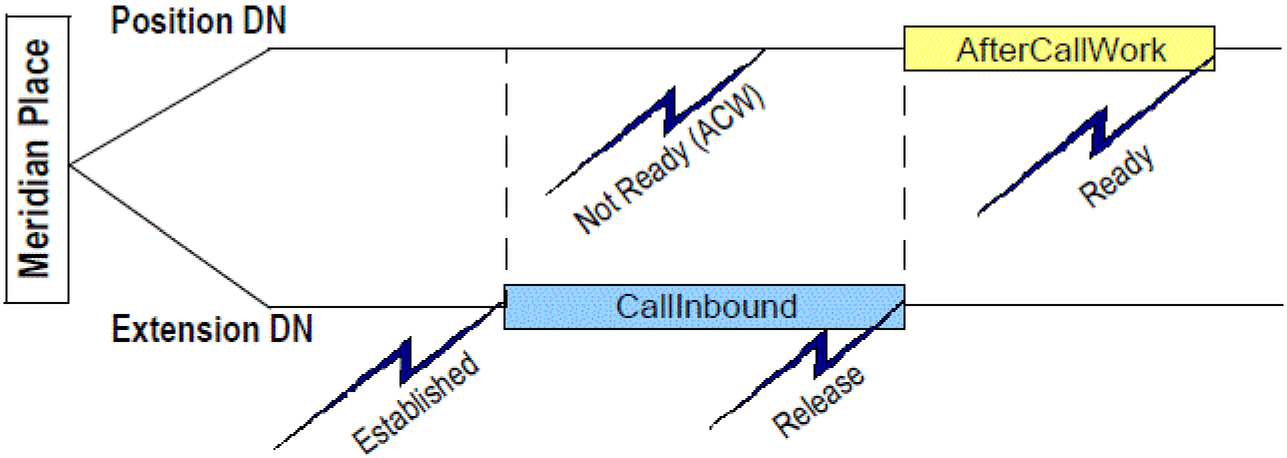
The diagram *ACW Request on a Position DN During a Telephony Interaction* illustrates this scenario when an inbound interaction is underway on the Position DN. An ACW-related event takes place during the interaction. Stat Server starts an `AfterCallWork` action when the interaction is released.



ACW Request on a Position DN During a Telephony Interaction

During the interaction on the Position DN, the agent presses the ACW button (workmode=ACW). Upon release of the interaction, Stat Server starts an `AfterCallWork` action on the Position DN. When Stat Server then receives the EventAgentReady TEvent, Stat Server terminates the `AfterCallWork` action. (`EventAgentLogout` and `EventAgentNotReady` with a workmode other than ACW would also terminate the action.)

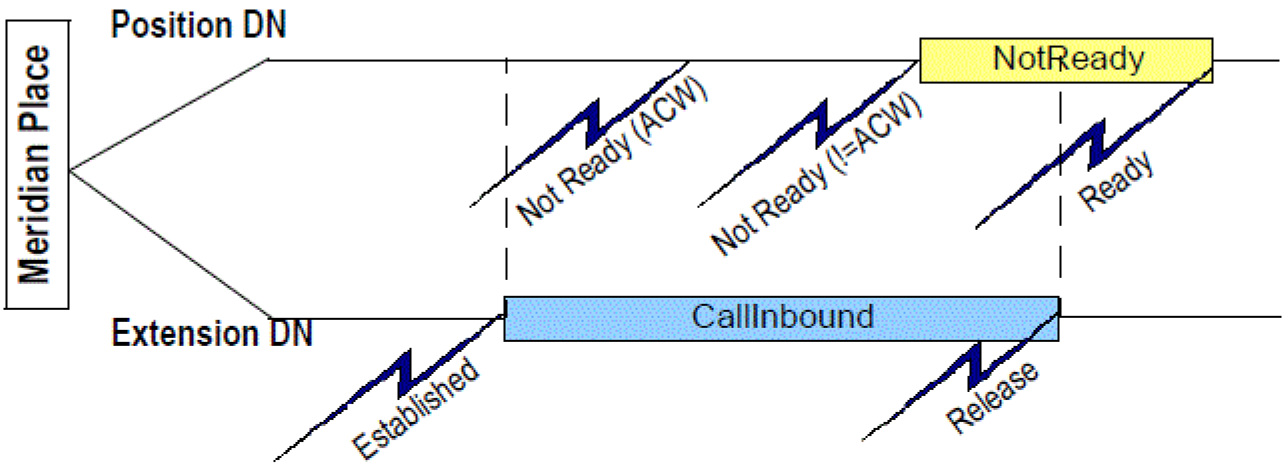
In the diagram *ACW Request on an Extension DN During a Telephony Interaction*, the agent presses the ACW button (workmode=ACW) while conducting an interaction on the Extension DN. Stat Server starts an `AfterCallWork` action *on the Position DN* upon release of the interaction, and terminates it under the same circumstances as those stated above. This termination occurs regardless of the DN from which the `AfterCallWork` action is generated.



ACW Request on an Extension DN During a Telephony Interaction

Clearing ACW During an Interaction

As a special provision, if, after having previously received an `EventAgentNotReady` TEvent (with workmode=ACW) while one or more calls are in progress on either the Position or Extension DN, Stat Server receives an `EventAgentReady` or `EventAgentNotReady` TEvent (with workmode!=ACW) while one or more calls are still in progress, Stat Server does not generate an `AfterCallWork` action upon release of the subsequent interaction(s).



Clearing an ACW Request During an Interaction

The diagram *Clearing an ACW Request During an Interaction* illustrates this scenario. It shows an interaction occurring on the Extension DN. During the interaction, the agent presses the ACW button. T-Server sends an `EventAgentNotReady` TEvent (workmode=ACW), and Stat Server registers it on the Position DN. Later, during the same interaction, the agent presses the `NotReady` button. T-Server sends an `EventAgentNotReady` TEvent (with workmode!=ACW), and Stat Server acknowledges it on the Position DN. As this TEvent and workmode combination terminate after-call work, Stat Server does not start an `AfterCallWork` action when the interaction terminates, but rather immediately starts a `NotReady` action on the Position DN when the `NotReady` button is pressed.

> Important
>
> This ACW model applies when Stat Server 7.0^{+} is used in conjunction with Meridian T-Server 7.0. Contact Genesys Customer Care to understand Stat Server 7.0 behavior if the version of your Meridian T-Server is less than 7.0.

Back to top

AgentActive

For Stat Server 8.1.0 and lower releases (8.1.0^{-}), this durable action starts on a mediation DN when the status of an agent, who is already logged into that mediation DN through a regular DN that belongs to a place, changes from `NotReadyForNextCall`. This action ends when agent status changes to `NotReadyForNextCall` on that mediation DN, when that agent logs out from the mediation DN, or when the `NotMonitored` action starts. For 8.1.2 and higher releases (8.1.2^{+}), Stat Server generates this action only on DNs on which there is a known agent (the assignment of those DNs to a place is not required).

> **Important**
>
> In order to use this action in statistics that are accessing action attributes (for example: filters, formulas, `GroupBy` , `DistinguishBy`) set the **[statserver]**/queue-use-pseudo-actions option to `false`.

Back to top

AgentLogin (Regular DNs)

This momentary action occurs when Stat Server detects agent login to a DN through either of the following:

- Stat Server receives `EventAgentLogin` on the DN.
- Stat Server receives `EventRegistered` or `EventAddressInfo` for the DN indicating agent login.

Stat Server generates this media-independent action when Stat Server detects login to the device, not to a particular media channel on the device.

For a description of this action on mediation DNs, see AgentLogin (Mediation DNs).

Back to top

AgentLogin (Mediation DNs)

For Stat Server 8.1.0 and lower (releases 8.1.0^{-}), this durable action starts when agent logs on to a mediation DN through a regular DN that belongs to a place and for which the agent is known. This action ends when the agent logs out from the mediation DN or when the `NotMonitored` action starts. For 8.1.2 (8.1.2^{+}) and higher release, Stat Server generates this action only on DNs for which there is a known agent (the assignment of those DNs to a place is not required).

> **Important**
>
> In order to use this action in statistics that are accessing action attributes (for example: filters, formulas, `GroupBy` , `DistinguishBy`) set the **[statserver]**/queue-use-pseudo-actions option to `false`.

For a description of this action on regular DNs, see AgentLogin (Regular DNs).

Back to top

AgentLogout

`EventAgentLogout` triggers this retrospective, instantaneous action. Furthermore, this action inherits

its attributes (such as Reasons) from this TEvent, which can be useful, for example, for tallying the number of agent logout actions that occurred during a particular time frame because of a particular Reason (using Reason-based filtering introduced in the 7.6 release).

The duration of this action coincides with the duration of the agent's login on the DN. Stat Server generates this media-independent action when Stat Server detects:

- `EventAgentLogout` on a device—not when the agent logs off of a particular media channel.
- `EventLinkDisconnected` on a regular logged-in DN.

For 8.1.2 and higher releases (8.1.2$^+$), Stat Server generates this action only on DNs for which there is a known agent.

Back to top

AgentReady

For Stat Server 8.1.0 and lower releases (8.1.0$^-$), this durable action starts on a mediation DN when the status of agent, who is already logged into that mediation DN through a regular DN belonging to a place, changes to `WaitForNextCall`. This action ends when agent status changes from `WaitForNextCall` on that mediation DN, when that agent logs out from the mediation DN, or when the `NotMonitored` action starts. (See Place and Agent Status for a definition of agent status). For 8.1.2 and higher releases (8.1.2$^+$), Stat Server generates this action only on DNs for which there is a known agent (the assignment of those DNs to a place is not required).

> **Important**
>
> In order to use this action in statistics that are accessing action attributes (for example: filters, formulas, `GroupBy` , `DistinguishBy`) set the **[statserver]**/queue-use-pseudo-actions option to `false`.

Back to top

ASM_Engaged

This durable action is specific to DNs of the `Extension` or `Position` type that are involved with the outbound predictive dialing, which runs in `Predictive with seizing` mode and is based on the Active Switching Matrix (ASM) call model.

This action starts upon Stat Server’s receipt of:

- `EventEstablished` on the communication port DN (CPDN).
- `EventEstablished` on the agent DN where its UserData attribute contains the <`"GSW_CALL_TYPE", "ENGAGING"`> key-value pair.

Prior to Stat Server 7.6, this action started upon receipt of `EventRinging`. Now, upon receiving `EventRinging` with `ANI/OtherDN` pointing to the CPDN, Stat Server generates the `CallRinging` action.

N-Dialer makes a predictive dialing call to a customer number and delivers an engaging call (of the Inbound or Internal type) to an agent via a CPDN. The action indicates that the agent on a particular DN is waiting for the customer to be connected.

This action ends for communication port DNs when any of the following occur:

- The ASM_Outbound action starts on the CPDN.
- The customer is connected to the agent.
- Either the predictive dialing or the engaging call is released (through receipt of EventReleased or EventAbandoned) before the agent and the customer are connected to each other.
- The NotMonitored action starts.

This action ends for agent DNs when any of the following occurs:

- The ASM_Outbound action starts on the agent DN.
- Either the predictive dialing or the engaging call is released (through receipt of EventReleased or EventAbandoned) before the agent and the customer are connected to each other.
- The NotMonitored action starts.

> **Tip**
>
> Refer to the *Outbound Contact Deployment Guide* for information on the ASM call model.

Back to top

ASM_Outbound

This durable action is specific to DNs of the Extension or Position type that are involved with the outbound predictive dialing, which runs in Predictive with seizing mode and is based on the Active Switching Matrix (ASM) call model.

This action starts upon Stat Server’s receipt of:

- EventAttachedDataChanged on the agent DN with UserData containing the (''GSW_RECORD_HANDLE'', <*any value*>) key-value pair.
- EventPartyChanged on the agent DN with PreviousConnID pointing to a call that Stat Server recognizes as ASM-engaged and UserData containing the <''GSW_CALL_TYPE'',''REGULAR''> key-value pair.

This action ends on the CPDN when either the agent or the customer releases the call or if the NotMonitored action starts. On the agent DN, this action ends when the call ends on the agent’s DN or when the NotMonitored action starts.

> **Tip**
>
> Refer to the *Outbound Contact Deployment Guide* for information on the ASM call model.

Back to top

Available

This durable action indicates that an agent (or place) is ready to receive interactions on a particular media channel. This action is similar to `WaitForNextCall` in the telephony model.

Back to top

BeingCoached

Stat Server generates this momentary action when coaching begins on a chat interaction, whether by invitation or not.

Back to top

BeingMonitored

Stat Server generates this momentary action when monitoring begins on a chat interaction.

Back to top

Blocked

This durable action indicates that an agent (or place) has put himself or herself into the `NotReady` state for a particular media, and/or that he or she has selected `DoNotDisturb`. This action is similar to the `NotReadyForNextCall` action.

Back to top

CallAbandoned

This retrospective action derives from the CallWait durable action if `CallWait` terminates because of `EventAbandoned` with an `AttributeReliability` attribute equal to `TReliabilityOk`.

`CallAbandoned` is always simultaneous with one of the following call-type actions:

- `CallAbandonedUnknown`
- `CallAbandonedInternal`
- `CallAbandonedInbound`

- CallAbandonedOutbound
- CallAbandonedConsult

The interaction type that Stat Server receives from T-Server with EventQueued or EventRouteRequest determines which of the above five actions occurs simultaneously with CallAbandoned.

Back to top

CallAbandonedFromDialing

This retrospective action derives from the CallDialing durable action if CallDialing terminates because of EventReleased and if the interaction is not a consult call with an interaction state of Transferred or Conferenced.

CallAbandonedFromDialing is always simultaneous with one of the following call-type actions:

- CallAbandonedFromDialingUnknown
- CallAbandonedFromDialingInternal
- CallAbandonedFromDialingInbound
- CallAbandonedFromDialingOutbound
- CallAbandonedFromDialingConsult

The interaction type that Stat Server receives from T-Server with EventReleased determines which of the above five actions occurs simultaneously with CallAbandonedFromDialing.

Back to top

CallAbandonedFromHold

This retrospective action derives from the CallOnHold durable action if CallOnHold terminates because of EventReleased with an interaction state other than Transferred.

CallAbandonedFromHold is always simultaneous with one of the following call-type actions:

- CallAbandonedFromHoldUnknown
- CallAbandonedFromHoldInternal
- CallAbandonedFromHoldInbound
- CallAbandonedFromHoldOutbound
- CallAbandonedFromHoldConsult

The interaction type that Stat Server receives from T-Server with EventReleased determines which of the above five actions occurs simultaneously with CallAbandonedFromHold.

Back to top

CallAbandonedFromRinging (Regular DNs)

This retrospective action derives from the CallRinging durable action if CallRinging terminates because of EventReleased or EventAbandoned (specifically, without interaction state 22-Redirected or 23-Forwarded). The AttributeReliability attribute, a new attribute provided with 7.1 T-Servers, must accompany EventAbandoned and this attribute's value must equal TReliabilityOk.

CallAbandonedFromRinging is always simultaneous with one of the following call-type actions:

- CallAbandonedFromRingingUnknown
- CallAbandonedFromRingingInternal
- CallAbandonedFromRingingInbound
- CallAbandonedFromRingingOutbound
- CallAbandonedFromRingingConsult

The interaction type that Stat Server receives from T-Server with EventReleased or EventAbandoned determines which of the above five actions occurs simultaneously with CallAbandonedFromRinging.

This action may occur simultaneously with the retrospective CallAbandonedFromRinging (Mediation DNs) action.

Back to top

CallAbandonedFromRinging (Mediation DNs)

For regular interactions, this retrospective action occurs at a mediation DN when EventReleased (with an interaction state other than CallForwarded or CallRedirected) is received after EventRinging from a DN to which an interaction was going to be distributed from the mediation DN. It receives as its duration the interval from the moment when the interaction entered the mediation DN (EventQueued or EventRouteRequest) to the moment when the interaction was abandoned (EventReleased).

For hunt-call interactions, this retrospective action occurs at a mediation DN when EventAbandoned is received on that DN, given that EventRinging had been previously received on at least one agent DN, belonging to a hunt group. The resultant action receives as its duration from the moment that the call entered the mediation DN (EventQueued or EventRouteRequest) to the moment when the interaction was abandoned (EventAbandoned).

Back to top

CallAbandonedFromRinging (Virtual Queues)

For virtual queue objects that are controlled by a Multimedia-monitored switch this retrospective action occurs when Stat Server receives from Interaction Server the EventRevoked event with the Abandoned reason.

The duration that Stat Server prescribes to this action is the interval from EventQueued to EventRevoked.

This action is similar to CallAbandonedFromRinging (Mediation DNs) in the telephony model.

Back to top

CallAnswered (Regular DNs)

This retrospective action derives from the CallRinging durable action if `CallRinging` terminates because of `EventEstablished`.

`CallAnswered` is always simultaneous with one of the following call-type actions:

- `CallAnsweredUnknown`
- `CallAnsweredInternal`
- `CallAnsweredInbound`
- `CallAnsweredOutbound`
- `CallAnsweredConsult`

The interaction type that Stat Server receives from T-Server with `EventEstablished` determines which of the above five actions occurs simultaneously with `CallAnswered (Regular DNs)`.

This action may occur simultaneously with the retrospective mediation DN action CallAnswered (Mediation DNs).

Back to top

CallAnswered (Mediation DNs)

This retrospective action occurs at a mediation DN when `EventEstablished` is received after `EventRinging` from a DN to which an interaction was distributed from the mediation DN. `CallAnswered` receives as its duration the interval from the moment when the interaction enters the mediation DN (the latest of the `EventQueued`, `EventRouteRequest` or `EventPartyChanged` TEvents if it occurs while the call is waiting in queue or at the routing point) to the moment when the agent takes the interaction (`EventEstablished` or `EventDiverted`, whichever is latest).

> **Important**
>
> If an interaction was accepted at an agent DN at moment T1 and the interaction is subsequently requeued to a mediation DN (at moment T2), Stat Server will not generate the `CallAnswered` action on all mediation DNs for which the `EventDiverted` or `EventRouteUsed` TEvents were delayed (that is, when these events follow T2).

`CallAnswered` is always simultaneous with one of the following call-type actions:

- `CallAnsweredUnknown`
- `CallAnsweredInternal`
- `CallAnsweredInbound`
- `CallAnsweredOutbound`

- `CallAnsweredConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with `CallAnswered (Mediation DNs)`.

This action may occur simultaneously with the CallAnswered (Regular DNs) retrospective action.

Back to top

CallAnswered (Virtual Queues)

For virtual queue objects that are controlled by a Multimedia-monitored switch this retrospective action occurs when Stat Server receives `EventPartyAdded` as a result of an agent accepting the interaction. The duration that Stat Server prescribes to this action is the interval from `EventQueued` to `EventPartyAdded`.

This action is similar to CallAnswered (Mediation DNs) in the telephony model.

Back to top

CallCleared

Stat Server generates this retrospective action only on a virtual queue. The action derives from the CallWait durable action if `CallWait` terminates because of `EventDiverted` with an interaction state of `Redirected`. With this event, the Universal Routing Server, by means of T-Server, indicates that an interaction has left this queue and is being delivered to an agent from another virtual queue.

`CallCleared` is always simultaneous with one of the following call-type actions:

- `CallAbandonedUnknown`
- `CallAbandonedInternal`
- `CallAbandonedInbound`
- `CallAbandonedOutbound`
- `CallAbandonedConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with `CallCleared`.

Back to top

CallConferenceJoined

This momentary action occurs in a conference call at a DN that was added to the conference. `CallConferenceJoined` derives from:

- `EventPartyChanged` for a two step conference

- EventEstablished for a single step conference

with an interaction state of Conferenced.

Back to top

CallConferenceMade

Once the transfer completes, this momentary action occurs at the DN that initiated the conference. CallConferenceMade derives from EventPartyAdded with an interaction state of Conferenced, a ThirdPartyDNRole of AddedBy, and a ThirdPartyDN equal to ThisDN.

Back to top

CallConferenceOriginated

This durable action measures the amount of time that an agent spent in a three-party conference. In regular PBX scenarios, this action starts when the originating agent invites another agent to a call (EventPartyChanged) and stops when the originating agent leaves the conference (EventReleased). The CallConferenceOriginated action is not supported in blind conferences when a conference completes while the call is at a routing point or ACD queue.

For CallConferenceOriginated actions that are triggered by the EventPartyChanged TEvent with the CallState attribute set to Conferenced, all attributes (ThisQueue, DNIS, and others) are now taken from this TEvent.

In network-attended transfer and conference scenarios, this action starts when Stat Server receives NetworkCallStateConferenced as the value of the AttributeNetworkCallState attribute for the originating agent and stops when this attribute’s value becomes NetworkCallStateReconnected, NetworkCallStateDisconnected, NetworkCallStateTransferred or NetworkCallStateConferenced for the originating agent or when the NotMonitored action starts.

> **Important**
>
> When specified in the MainMask of a stat type, Stat Server ignores DistByConnID Formula assignments, since, by definition, this action may occur only once for a given connection ID.

Statistics based on this action include the originating agent's continued involvement in conferenced calls, regardless of whether this involvement is active or inactive.

> **Important**
>
> Using this action to measure the originating agent’s time in a three-party conference presumes that the originating agent leaves the conference first. If the customer or the conferenced-in agent leaves the conference, Stat Server continues to tally this metric until the originating agent leaves the transaction.

Back to top

CallConferencePartyAdded

This momentary action occurs at all DNs participating in a conference call when a new DN joins the conference. `CallConferencePartyAdded` derives from `EventPartyAdded` with a `ThirdPartyDNRole` of `AddedBy` and a `ThirdPartyDN` different from `ThisDN`.

Back to top

CallConferencePartyDeleted

This momentary action occurs in a conference call at all DNs left in the conference when a DN ends its participation in the conference. It derives from `EventPartyDeleted`.

Back to top

CallConsult

This durable action starts when Stat Server receives `EventEstablished` from a DN with a value of `Consult` for the interaction-type parameter. Call origination, whether from within the contact center or outside, is not indicated. This action's corresponding initial momentary action is `CallConsultStarted`.

`CallConsult` ends with `EventReleased` or `EventPartyChanged` for the same call or when the `NotMonitored` action starts. When `CallConsult` ends with `EventReleased`, it causes the `CallConsultCompleted` retrospective action to occur. When `CallConsult` ends with `EventPartyChanged`, it causes the `CallPartyChanged` retrospective action to occur.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallConsultCompleted

This retrospective action derives from the `CallConsult` durable action. `CallConsultCompleted` is generated when a consultation call completes.

Use `CallConsultCompleted` instead of `CallConsult` for filtering attached data at the end of actions.

Back to top

CallConsultOriginated

This durable action starts when Stat Server receives EventEstablished from a DN with a value of Consult for the interaction-type parameter. This action is similar to a CallConsult action providing additional information about call origination–namely, from an agent's DN. Its corresponding initial momentary action is CallConsultStarted.

CallConsultOriginated ends with EventReleased or EventPartyChanged for the same call or when the NotMonitored action starts. When CallConsultOriginated ends with EventPartyChanged, this action causes Stat Server to generate the CallPartyChanged retrospective action.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallConsultReceived

This durable action starts when Stat Server receives EventEstablished from a DN with a value of Consult for the interaction-type parameter. This action is similar to a CallConsult action providing additional information about call origination—namely, from a DN outside the contact center. Its corresponding initial momentary action is CallConsultStarted.

CallConsultReceived ends with EventReleased or EventPartyChanged for the same call or when the NotMonitored action starts. When it ends with EventPartyChanged, it causes the retrospective action CallPartyChanged.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallConsultStarted

This momentary action occurs whenever the CallConsult, CallConsultOriginated, or CallConsultReceived durable action starts.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallDialConferenced

This retrospective action derives from the CallDialing durable action if CallDialing terminates because of EventReleased for a consult call with an interaction state of Conferenced. CallDialConferenced is interaction-type specific, so it can also be considered to derive from CallDialingConsult.

Back to top

CallDialTransferred

This retrospective action derives from the CallDialing durable action if CallDialing terminates because of EventReleased for a consult call with an interaction state of Transferred. CallDialTransferred is interaction-type specific, so it can also be considered to derive from CallDialingConsult.

Back to top

CallDialed

This retrospective action derives from the CallDialing durable action if CallDialing terminates because of EventEstablished.

CallDialed is always simultaneous with one of the following call-type actions:

- CallDialedUnknown
- CallDialedInternal
- CallDialedInbound
- CallDialedOutbound
- CallDialedConsult

The interaction type that Stat Server receives from T-Server with EventDialing determines which of the above five actions occurs simultaneously with CallDialed.

Back to top

CallDialing

This durable action starts when Stat Server receives EventDialing from T-Server for a DN. Its corresponding initial momentary action is CallDialingStarted.

This action lasts until Stat Server receives either EventEstablished or EventReleased for the same call, or until the NotMonitored action starts. If EventEstablished or EventReleased is received, and, in the latter case, if the interaction is a consult call with a call state of Transferred or Conferenced, the termination of CallDialing produces the retrospective action CallDialed, CallAbandonedFromDialing, CallDialTransferred, or CallDialConferenced.

CallDialing is always simultaneous with one of the following call-type actions:

- CallDialingUnknown
- CallDialingInternal
- CallDialingInbound
- CallDialingOutbound
- CallDialingConsult

The interaction type that Stat Server receives from T-Server with EventDialing determines which of the preceding five actions occurs simultaneously with CallDialing.

Back to top

CallDialingStarted

This momentary action occurs whenever the CallDialing durable action starts.

CallDialingStarted is always simultaneous with one of the following call-type actions:

- CallDialingStartedUnknown
- CallDialingStartedInternal
- CallDialingStartedInbound
- CallDialingStartedOutbound
- CallDialingStartedConsult

The interaction type that Stat Server receives from T-Server with EventDialing determines which of the above five actions occurs simultaneously with CallDialingStarted.

Back to top

CallDistributed

This retrospective action derives from the CallWait durable action if CallWait terminates because Stat Server receives EventRouteUsed or EventDiverted from T-Server.

In addition, for virtual queue objects, EventDiverted must contain an interaction state other than Redirected.

For T-Server-originating events, CallDistributed is always simultaneous with one of the following call-type actions:

- CallDistributedUnknown
- CallDistributedInternal
- CallDistributedInbound
- CallDistributedOutbound

- `CallDistributedConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with `CallDistributed`.

Back to top

CallDistributedToQueue

Stat Server generates this retrospective action on a mediation DN (`DN1`) if an interaction is distributed from this DN to a second mediation DN (`DN2`). The duration of this action is equal to the time from receipt of an `EventQueued` or `EventRouteRequest` TEvent on `DN1` until the receipt of an `EventQueued` or `EventRouteRequest` on `DN2`. Stat Server does not generate this action if an interaction enters `DN2` but has not been distributed from `DN1`. Stat Server also does not generate this action if an interaction is distributed from `DN1` to a nonmediation DN, such as to an agent's DN. After Stat Server generates `CallDistributedToQueue` on `DN1`, `DN1` is cleared from the list of DNs from which the interaction can be distributed.

`CallDistributedToQueue` is always simultaneous with one of the following call-type actions:

- `CallDistributedToQueueInternal`
- `CallDistributedToQueueInbound`
- `CallDistributedToQueueOutbound`
- `CallDistributedToQueueConsult`
- `CallDistributedToQueueUnknown`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with `CallDistributedToQueue`.

Back to top

CallEntered

This momentary action occurs, depending on the type of the DN, when Stat Server receives `EventQueued` or `EventRouteRequest` from T-Server.

For T-Server-originating events, `CallEntered` is always simultaneous with one of the following call-type actions:

- `CallEnteredUnknown`
- `CallEnteredInternal`
- `CallEnteredInbound`
- `CallEnteredOutbound`
- `CallEnteredConsult`

The interaction type that Stat Server receives from T-Server with EventQueued or EventRouteRequest determines which of the above five actions occurs simultaneously with CallEntered.

> **Important**
>
> CallEntered will be generated only once for the queue and the same still active call.

Back to top

CallForwarded (Regular DNs)

This retrospective action derives from the CallRinging durable action if CallRinging terminates because of EventReleased with an interaction state of Forwarded or Redirected (when the forwarding functionality is enabled on a DN).

CallForwarded is always simultaneous with one of the following call-type actions:

- CallForwardedUnknown
- CallForwardedInternal
- CallForwardedInbound
- CallForwardedOutbound
- CallForwardedConsult

This action may occur simultaneously with the CallForwarded (Mediation DNs) retrospective action.

Back to top

CallForwarded (Mediation DNs)

For regular interactions, this retrospective action occurs at a mediation DN when Stat Server receives EventReleased (with an interaction state of CallForwarded or CallRedirected) following EventRinging from a DN to which an interaction was going to be distributed from the mediation DN. Action duration is the interval from the moment when the interaction enters the mediation DN (EventQueued or EventRouteRequest) to the moment when the interaction is abandoned (EventReleased).

For hunt-call interactions, Stat Server never generates this action.

Back to top

CallHeld

This momentary action occurs whenever the CallOnHold durable action starts.

CallHeld is always simultaneous with one of the following call-type actions:

- CallHeldUnknown
- CallHeldInternal
- CallHeldInbound
- CallHeldOutbound
- CallHeldConsult

The interaction type that Stat Server receives from T-Server with EventHeld determines which of the above five actions occurs simultaneously with CallHeld.

Back to top

CallInbound

This durable action starts when Stat Server receives:

- EventEstablished.
- EventPartyChanged from a DN with a value of Inbound for the interactiontype parameter.

Its corresponding initial momentary action, upon receipt of EventEstablished, is CallInboundStarted. Stat Server generates this action upon receipt of EventPartyChanged when T-Server configuration causes T-Server to transmit an Inbound interaction type with the TEvent rather than Consult. This can happen, for example, when the use-data-from T-Server configuration option is set to consult-user-data.

CallInbound ends with EventReleased for the same interaction, causing the CallInboundCompleted retrospective action to occur, when EventPartyChanged is received for a different party, or when the NotMonitored action starts.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallInboundCompleted

This retrospective action derives from the CallInbound durable action. CallInboundCompleted is generated when an inbound interaction completes.

Use CallInboundCompleted instead of CallInbound for filtering attached data at the end of actions.

Back to top

CallInboundStarted

This momentary action occurs whenever the `CallInbound` durable action starts.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallInternal

This durable action starts when Stat Server receives `EventEstablished` from a DN with a value of `Internal` for the interaction-type parameter. Its corresponding initial momentary action is `CallInternalStarted`.

`CallInternal` ends with `EventReleased` for the same interaction, causing the `CallInternalCompleted` retrospective action to occur, or when the `NotMonitored` action starts.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallInternalCompleted

This retrospective action derives from the `CallInternal` durable action. `CallInternalCompleted` is generated when an internal interaction completes.

Use `CallInternalCompleted` instead of `CallInternal` for filtering attached data at the end of actions.

Back to top

CallInternalOriginated

This durable action starts when Stat Server receives `EventEstablished` from a DN with a value of `Internal` for the interaction-type parameter. This action is similar to a `CallInternal` action, providing additional information about interaction origination—namely, from an agent’s DN. Its corresponding initial momentary action is `CallInternalStarted`.

`CallInternalOriginated` ends with `EventReleased` for the same interaction or when the `NotMonitored` action starts.

> Tip
> See also DN Actions at Newly Registered DNs.

Back to top

CallInternalReceived

This durable action starts when Stat Server receives `EventEstablished` from a DN with a value of `Internal` for the interaction-type parameter. This action is similar to a `CallInternal` action, providing additional information about origination of the interaction—namely, from a DN not belonging to the agent. Its corresponding initial momentary action is `CallInternalStarted`.

`CallInternalReceived` ends with `EventReleased` for the same interaction or when the `NotMonitored` action starts.

> Tip
> See also DN Actions at Newly Registered DNs.

Back to top

CallInternalStarted

This momentary action occurs whenever the `CallInternal` durable action starts.

> Tip
> See also DN Actions at Newly Registered DNs.

Back to top

CallMissed

This retrospective action occurs on a mediation DN when `EventReleased` comes after `EventEstablished`. It applies to calls distributed from a source other than the mediation DN, on which the agent is logged in. Action duration is the interval beginning with `EventEstabished` and ending with `EventReleased`.

Action `CallMissed` is not generated on a mediation DN at the time when a call is released on an agent's DN, if at that moment the agent's DN is no longer associated with the mediation DN, either through an agent group or place group.

Back to top

CallObserved...

The CallObserved... actions include the following:

- CallObservedUnknown
- CallObservedInternal
- CallObservedInbound
- CallObservedOutbound
- CallObservedConsult

One of these durable actions starts in one the following cases:

- Stat Server receives EventPartyAdded with ThisDNRole equal to Destination and OtherDNRole equal to Observer.
- For ConnID Stat Server receives the EventPartyChanged event with PreviousConnID not equal to ConnID and action CallObserved... existed for the PreviousConnID.

The action terminates in one the following cases:

- Stat Server receives EventPartyDeleted for the agent's DN with OtherDNRole equal to Observer.
- Stat Server receives EventReleased for the interaction.
- For PreviousConnID Stat Server receives the EventPartyChanged event with PreviousConnID not equal to ConnID.
- The NotMonitored action starts.

Supervisor participation in an interaction does not affect the Service Observed statistics.

> ### Tip
> For information on the T-Server call model, refer to the **Service Observing an Agent** section in the T-Library SDK C Developer's Guide. See also DN Actions at Newly Registered DNs.

Back to top

CallObserving...

The CallObserving... actions include the following:

- CallObservingUnknown
- CallObservingInternal

- CallObservingInbound
- CallObservingOutbound
- CallObservingConsult

One of these durable actions starts in one the following cases:

- Stat Server receives the EventEstablished event with ThisDNRole equal to RoleObserver and CallState equal to CallStateBridged.
- For ConnID Stat Server receives the EventPartyChanged event with PreviousConnID not equal to ConnID and action CallObserving... existed for the PreviousConnID.

The action terminates in one the following cases:

- Stat Server receives the EventReleased event.
- For PreviousConnID Stat Server receives the EventPartyChanged event with PreviousConnID not equal to ConnID.
- The NotMonitored action starts.

> Tip
>
> Observer actions can be used in statistics with DistByConnID in formulas.

This action was introduced in release 8.5.1.

Back to top

CallOnHold

This durable action starts when Stat Server receives EventHeld from T-Server for a DN. Its initial momentary action is CallHeld.

This action lasts until Stat Server receives either EventRetrieved or Event Released for the same interaction, or until the NotMonitored action starts. If Stat Server receives EventRetrieved or EventReleased and, in the latter case, if the interaction state is Transferred, termination of CallOnHold produces one of the following retrospective actions:

- CallRetrievedFromHold
- TransferredFromHold
- CallAbandonedFromHold

CallOnHold is always simultaneous with one of the following call-type actions:

- CallOnHoldUnknown
- CallOnHoldInternal
- CallOnHoldInbound

- CallOnHoldOutbound
- CallOnHoldConsult

The interaction type that Stat Server receives from T-Server with EventHeld determines which of the above five actions occurs simultaneously with CallOnHold.

When determining status, Stat Server temporarily hides from consideration the corresponding DN action (CallInternal, CallInbound, CallOutbound, or CallUnknown) of an established telephony interaction on the same DN for the duration that the interaction is on hold.

Back to top

CallOutbound

This durable action starts when Stat Server receives EventEstablished from a DN with a value of Outbound for the interaction-type parameter. Its corresponding initial momentary action is CallOutboundStarted.

CallOutbound ends with EventReleased for the same interaction, causing the CallOutboundCompleted retrospective action to occur, or when the NotMonitored action starts.

> Tip
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallOutboundCompleted

This retrospective action derives from the CallOutbound durable action. CallOutboundCompleted is generated when an outbound interaction completes.

Use CallOutboundCompleted instead of CallOutbound for filtering attached data at the end of actions.

Back to top

CallOutboundOriginated

This durable call-related action is started when the EventEstablished event is received for a given ConnID and there exists the CallDialing action for that ConnID. This action ends with the EventReleased event.

> Tip

> The CallOutboundOriginated action can be used in statistics with DistByConnID in formulas.

This action was introduced in release 8.5.1.

Back to top

CallOutboundReceived

This durable call-related action is started when the EventEstablished event is received for a given ConnID and there exists the CallRinging action for that ConnID. This action ends with the EventReleased event.

> **Tip**
>
> The CallOutboundReceived action can be used in statistics with DistByConnID in formulas.

This action was introduced in release 8.5.1.

Back to top

CallOutboundStarted

This momentary action occurs whenever the CallOutbound durable action starts.

> **Tip**
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallPartyChanged

This retrospective action derives from the following:

- The CallConsult, the CallConsultOriginated, or the CallConsultReceived durable actions if any of these actions terminate because of EventPartyChanged.
- The CallInbound action, in circumstances where T-Server configuration causes T-Server to transmit an Inbound interaction type with the TEvent rather than Consult, such as may be the case when the use-data-from T-Server configuration option is set to consult-user-data.

Back to top

CallReleased (Mediation DNs)

This retrospective action occurs at a mediation DN when `EventReleased` comes after `EventEstablished` from a regular DN, for an interaction distributed from the mediation DN. Action duration is the interval from `EventEstabished` to `EventReleased`.

Back to top

CallReleased (Virtual Queues)

For virtual queue objects that are controlled by a Multimedia-monitored switch this retrospective action occurs when Stat Server receives `EventPartyRemoved` as a result of an agent finishing an interaction. Stat Server calculates the duration from the moment of acceptance of an interaction (`EventPartyAdded`) until the moment that the last involved party of the interaction leaves it (`EventPartyRemoved`).

Stat Server does not generate this action if an interaction is offered to a contact-center handling resource but the resource does not explicitly accept or answer it. Such may be the case where the configured time interval for acceptance times out and Stat Server receives the EventRevoked event from Interaction Server.

This action is similar to CallReleased (Mediation DNs) in the telephony model.

Back to top

CallRetrievedFromHold

This retrospective action derives from the `CallOnHold` durable action if `CallOnHold` terminates because of `EventRetrieved`.

`CallRetrievedFromHold` is always simultaneous with one of the following call-type actions:

- `RetrievedFromHoldUnknown`
- `RetrievedFromHoldInternal`
- `RetrievedFromHoldInbound`
- `RetrievedFromHoldOutbound`
- `RetrievedFromHoldConsult`

The interaction type that Stat Server receives from T-Server with `EventEstablished` determines which of the above five actions occurs simultaneously with `CallRetrievedFromHold`.

Back to top

CallRinging

This durable action starts when Stat Server receives either:

- `EventRinging` from T-Server for a DN or, for an interaction derived from a consult call, when Stat Server receives `EventPartyChanged`.
- `EventPartyChanged` in circumstances where T-Server configuration causes T-Server to transmit an `Inbound` interaction type with the TEvent rather than `Consult`, such as may be the case when the `use-data-from` T-Server configuration option is set to `consult-user-data`.

Its initial momentary action is CallRingingStarted. `CallRinging` lasts until Stat Server receives:

- `EventEstablished`
- `EventReleased`
- `EventPartyChanged` for a consult call and for the same interaction

Or, until the NotMonitored (Regular DNs) action starts.

If `EventEstablished`, `EventReleased`, or, for a consult call, `EventPartyChanged` is received, the termination of `CallRinging` produces the retrospective action CallAnswered (Regular DNs), CallAbandonedFromRinging (Regular DNs) , or CallRingingPartyChanged (Regular DNs) .

`CallRinging` is always simultaneous with one of the following call-type actions:

- `CallRingingUnknown`
- `CallRingingInternal`
- `CallRingingInbound`
- `CallRingingOutbound`
- `CallRingingConsult`

The interaction type that Stat Server receives from T-Server with `EventRinging` determines which of the above five actions occurs simultaneously with `CallRinging`.

Back to top

CallRingingPartyChanged (Regular DNs)

This retrospective action derives from the following:

- The `CallRinging` durable action, if `CallRinging` terminates because of `EventPartyChanged` for a consult call.
- The `CallRingingConsult` action, as `CallRingingPartyChanged (Regular DNs)` is interaction-type-specific.
- The `CallInbound` action, in circumstances where T-Server configuration causes T-Server to transmit an `Inbound` interaction type with the TEvent instead of `Consult`, such as may be the case when the `use-data-from` T-Server configuration option is set to `consult-user-data`.

Back to top

CallRingingPartyChanged (Mediation DNs)

CallRingingPartyChanged (Mediation DNs) is a retrospective, interaction-related action reflecting Regular DN actions that Stat Server generates on Mediation DNs. Similar retrospective action is CallRingingPartyChanged (Regular DNs).

Back to top

CallRingingStarted

This momentary action occurs whenever the CallRinging durable action starts.

CallRingingStarted is always simultaneous with one of the following call-type actions:

- CallRingingStartedUnknown
- CallRingingStartedInternal
- CallRingingStartedInbound
- CallRingingStartedOutbound
- CallRingingStartedConsult

The interaction type that Stat Server receives from T-Server with EventRinging determines which of the above five actions occurs simultaneously with CallRingingStarted.

Back to top

CallTransferMade

This momentary action occurs at the DN from which a transfer was initiated (by TInitiateTransfer, TSingleStepTransfer, or TMuteTransfer, or by TMergeCalls) once the transfer is completed (EventReleased is received with an interaction state of Transferred).

CallTransferMade is always simultaneous with one of the following call-type actions:

- CallTransferMadeUnknown
- CallTransferMadeInternal
- CallTransferMadeInbound
- CallTransferMadeOutbound
- CallTransferMadeConsult

Back to top

CallTransferPartyChanged

Once the transfer completes, this momentary action occurs at the DN of the first party for a call transferred from a second party to a third. CallTransferPartyChanged derives from EventPartyChanged with an interaction state of Transferred and a ConnID equal to

PreviousConnID.

Back to top

CallTransferTaken

This momentary action occurs at the DN when a transfer is made, once the transfer completes (EventEstablished). This action requires one of the following conditions:

- Stat Server receives EventPartyChanged with an interaction state of Transferred and a ConnID different from PreviousConnID.
- Stat Server receives EventPartyChanged for this interaction on some mediation DN prior to distribution to a regular DN.
- Stat Server receives EventRinging with an interaction state of Transferred. (Refer to the description of the generate-transfer-taken-on-ringing configuration option in the Framework Stat Server Deployment Guide to learn how to control this aspect of CallTransferTaken action generation.)
- Stat Server receives EventRouteRequest with a CallState attribute of OK on a routing point if such event was preceded by EventQueued on the same routing point with a CallState attribute of Transferred.
- Note, that EventQueued will only be handled on a routing point, if the rp-handle-queueing-events configuration option in the [statserver] section has been set to true. (Refer to the option description in the Framework Stat Server Deployment Guide to learn how to control this aspect of CallTransferTaken action generation.)

> **Important**
>
> Stat Server counts transfers that are initiated from an agent's DN and completed on a queue or routepoint as TransferTaken for the agent receiving this call. In 7.x and lower releases, transfers initiated by an IVR were also counted as TransferTaken.

Back to top

CallTreatmentCompleted

This retrospective action is not derived from a durable action. CallTreatmentCompleted occurs when Stat Server receives EventTreatmentCompleted from T-Server, and the duration of this action is the total duration of the treatment.

> **Important**
>
> Stat Server handles treatment-related events only for Routing Points. In order to generate an appropriate action, a call with ConnID specified in the associated event should currently be waiting on a Routing Point.

Back to top

CallTreatmentNotStarted

This momentary action occurs when Stat Server receives `EventTreatmentNotApplied` from T-Server.

> **Important**
>
> Stat Server handles treatment-related events only for Routing Points. In order to generate an appropriate action, a call with `ConnID` specified in the associated event should currently be waiting on a Routing Point.

Back to top

CallTreatmentStarted

This momentary action occurs when Stat Server receives `EventTreatmentApplied` from T-Server.

> **Important**
>
> Stat Server handles treatment-related events only for Routing Points. In order to generate an appropriate action, a call with `ConnID` specified in the associated event should currently be waiting on a Routing Point.

Back to top

CallUnknown

This durable action starts when Stat Server receives `EventEstablished` from a DN with a value of Unknown for the interaction-type parameter. Its corresponding initial momentary action is CallUnknownStarted.

`CallUnknown` ends with `EventReleased` for the same interaction, causing the CallUnknownCompleted retrospective action to occur, or when the NotMonitored action starts.

> **Tip**
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallUnknownCompleted

This retrospective action derives from the CallUnknown durable action. `CallUnknownCompleted` is generated when an unknown interaction completes.

Use `CallUnknownCompleted` instead of `CallUnknown` for filtering attached data at the end of actions.

Back to top

CallUnknownStarted

This momentary action occurs whenever the CallUnknown durable action starts.

> **Tip**
>
> See also DN Actions at Newly Registered DNs.

Back to top

CallWait

Stat Server generates this durable action depending on the object’s type:

- Upon receipt of `EventQueued` (for ACD and virtual queue objects).
- Upon receipt of `EventRouteRequest` (for routing points objects).

Its corresponding initial momentary action is CallEntered.

`CallWait` action ends:

- Upon receipt of the following TEvents (for routing points and ACD and virtual queue objects):
  - `EventRouteUsed`
  - `EventDiverted`
  - `EventAbandoned`
  - `EventPartyChanged`
  - `EventReleased`
- Upon receipt of `EventAddressInfo` (for queue and routing point objects)
- When the `NotMonitored` action starts (such as when T-Server disconnects)

For T-Server-originating events, `CallWait` is always simultaneous with one of the following call-type actions:

- `CallWaitUnknown`

- `CallWaitInternal`
- `CallWaitInbound`
- `CallWaitOutbound`
- `CallWaitConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with CallWait.

Back to top

CoachingByIntrusionInitiated

This momentary action indicates that a resource has begun coaching a chat interaction without the invitation of the agent who is conducting the chat session.

Back to top

CoachingByRequestInitiated

This retrospective action indicates that an agent has begun coaching an interaction and not the initiator of this activity. Stat Server calculates the duration from the moment when coaching is started to the moment when coaching is finished.

This retrospective action is not derived from a durable action.

Back to top

CoachingRequested

This momentary action indicates that an agent requested coaching regardless of whether a coaching session was actually granted.

Back to top

ConferenceJoined

This momentary action, also called `InteractionConferenceJoined`, indicates that an agent has accepted and joined a conference. This action is similar to `CallConferenceJoined` in the telephony model.

Back to top

ConferenceJoinedByIntrusion

Stat Server generates this momentary action when a resource joins a conference without the invitation from the agent who is conducting the conference.

Back to top

ConferenceMade

This momentary action, also called `InteractionConferenceMade`, indicates that an agent has initiated a conference. This action is similar to `CallConferenceMade` in the telephony model.

Back to top

DND

This durable action starts on `RegDN` when Stat Server receives:

- The `EventRegistered` or `EventAddressInfo` TEvent with the `("dnd",1)` pair in the `Extensions` attribute
- The `EventDNDOn` TEvent

Previously started `DND` action ends when Stat Server receives one of the following TEvents:

- EventDNDOff
- EventServerDisconnected
- EventLinkDisconnected
- EventOutOfService
- EventAddressInfo with the `("dnd",0)` pair in the `Extensions` attribute

> ### Tip
>
> - The `DND` action is ignored in status calculations.
> - The `DND` action is media-independent (on a multi-media DN there can be only one instance of `DND`).
> - The `DND` action is unique on a given DN.

This action was introduced in release 8.5.107.

Back to top

Delivering

Stat Server generates this durable action, also called `InteractionDelivering`, for all interactions in the `Delivering` phase for a particular media on agent and/or place objects. `Delivering` follows `EventInvite`, and precedes receipt of `EventPartyChanged`, `EventRevoked`, and `EventRejected` for a particular interaction, agent, and media. This action is similar to CallRinging in the telephony model.

Back to top

DeliveringStarted

This momentary action, also called `InteractionDeliveringStarted`, marks the onset of interaction delivery (`Delivering`) for any interaction type, and it occurs when an agent is invited to an interaction. This action is similar to `RingingStarted` in the telephony model.

Back to top

DNActive

This durable action starts on a mediation DN when the status of regular DN, that is already logged in to that mediation DN changes from `NotReadyForNextCall`. This action ends when the regular DN's status changes to `NotReadyForNextCall`, when that regular DN logs out from the mediation DN, or when the NotMonitored action starts.

Back to top

DNLogin

This durable action starts on a mediation DN when a regular DN logs into the mediation DN. This action ends when that regular DN logs out from mediation DN or when the NotMonitored action starts.

Back to top

DNReady

This durable action starts on a mediation DN when the status of regular DN, already logged into that mediation DN, becomes `WaitForNextCall`.

This action ends on mediation DN when the status of regular DN stops being `WaitForNextCall`, when that regular DN logs out from the corresponding mediation DN, or when the NotMonitored action starts.

The counter that this action designates equals the number of DNs that are currently logged in to the queue when these DNs are in the WaitForNextCall status. That number does not include DN positions for Meridian and Meridian-like switches, for which associated extension DNs are not in WaitForNextCall status. (See Regular DN Status for a definition of DN status.)

Back to top

DoNotDisturb

Durable `DoNotDisturb` action for Agent/Place starts and ends upon relevant events simultaneously for every media-channel of a particular agent and associated by login the corresponding place. At any moment of time only one `DoNotDisturb` action can exist on a specific media-channel for the particular agent or place.

The DoNotDisturb action starts when Stat Server receives one of the following Interaction Server events:

- EventDonotDisturbOn
- EventAgentLogin (if the DND flag set to 1)
- EventPlaceAgentState (if the DND flag set to 1)
- EventMediaAdded (if an agent is currently in the DoNotDisturb state)

Previously started DoNotDisturb action ends when Stat Server receives one of the following Interaction Server events:

- EventDonotDisturbOff
- EventAgentLogout

> Tip
>
> - An agent can be logged in without logging into any media channel.
> - The DoNotDisturb action can be started/ended many times on a particular media-channel during the agent login session.
> - The DoNotDisturb action does not participate in the media-channel status calculation. However, if the media-channel status is Active or Available, Stat Server changes it to Blocked upon receiving the EventDoNotDisturbOn event from Interaction Server.

This action was introduced in release 8.5.110 .18.

Back to top

ExternalServiceRequested (Tenants)

This momentary action is generated on a Tenant upon receiving the EventExternalServiceRequested event.

Mandatory event attribute attr_actor_tenant_id specifies the Tenant's dbid.

The ExternalServiceRequested (Tenants) action supports the RequestEnvelope system attribute, associated with the attr_esp_request_envelope event attribute.

> Important
>
> The only supported Subject for the ExternalServiceRequested (Tenants) action is DNAction.

This action was introduced in release 8.5.104.

Back to top

ExternalServiceResponded (Tenants)

This momentary action is generated on a Tenant upon receiving the `EventExternalServiceResponded` event.

Mandatory event attribute `attr_actor_tenant_id` specifies the Tenant's dbid.

The `ExternalServiceResponded (Tenants)` action supports the `RequestEnvelope` system attribute, associated with the `attr_esp_request_envelope` event attribute.

> **Important**
>
> The only supported `Subject` for the `ExternalServiceResponded (Tenants)` action is `DNAction`.

This action was introduced in release 8.5.104.

Back to top

Handling

Stat Server generates this durable action, also called `InteractionHandling`, when an agent (or place) accepts an inbound, outbound, or internal interaction on a particular media. This action follows `EventPartyAdded` with `attr_party_type = 2` and has no equivalent in the telephony model. This action terminates when the agent leaves the interaction or when the `NotMonitored` action starts.

`Handling` is always simultaneous with one of the following interaction-type actions:

- `HandlingInbound`
- `HandlingInternal`
- `HandlingOutbound`

The interaction type that Stat Server receives from Interaction Server with `EventPartyAdded` and `attr_party_type = 2` determines which one of the above three actions occurs simultaneously with Handling.

> **Tip**
>
> Starting with Release 8.5.104, new ApplyFilterAtActionEndOnly stat type option is introduced, which can be used as additional filtering for the `Handling` action. See

> example below.

For example.
The stat type:

```
Category=TotalTime
MainMask=InteractionHandling
Objects=Agent, GroupAgents
Subject=DNAction
ApplyFilterAtActionEndOnly=yes
```

applied with `filter = PairExists( "CustomerSegment", "Gold" )`

The table below illustrates how the stat type above behaves with and without the ApplyFilterAtActionEndOnly specifier in a specific scenario:

Event	ApplyFilterAtActionEndOnly=no	ApplyFilterAtActionEndOnly=yes
T1: Interaction handling starts, `CustomerSegment=Gold`		
T2: Interaction data changed, `CustomerSegment=Silver`	The value is incremented by (**T2-T1**)	
T3: Interaction data changed, CustomerSegment=Gold		
T4: Interaction handling ends, CustomerSegment=Gold	The value is incremented by (**T4-T3**)	The value is incremented by (**T4-T1**)

Back to top

HandlingInbound

Stat Server generates this durable action, also called `InteractionHandlingInbound`, when an agent (or place) accepts an inbound interaction on a particular media. This action terminates when the agent leaves the interaction or when the `NotMonitored` action starts. HandlingInbound is similar to `CallInbound` in the telephony model.

Back to top

HandlingInboundStarted

Stat Server generates this momentary action, also called `InteractionHandlingInboundStarted`, when an agent accepts an inbound interaction. `HandlingInboundStarted` is similar to `CallInboundStarted` in the telephony model.

Back to top

HandlingInternal

Stat Server generates this durable action, also called `InteractionHandlingInternal`, when an agent (or place) accepts an internal interaction on a particular media. This action terminates when the agent leaves the interaction or when the `NotMonitored` action starts. `HandlingInternal` and is similar to `CallInternal` in the telephony model.

Back to top

HandlingInternalStarted

Stat Server generates this momentary action, also called `InteractionHandlingInternalStarted`, when an agent accepts an internal interaction. `HandlingInternalStarted` is similar to `CallInternalStarted` in the telephony model.

Back to top

HandlingOutbound

Stat Server generates this durable action, also called `InteractionHandlingOutbound`, when an agent (or place) accepts an outbound interaction on a particular media. This action terminates when the agent leaves the interaction or when the `NotMonitored` action starts. `HandlingOutbound` and is similar to `CallOutbound` in the telephony model.

Back to top

HandlingOutboundStarted

Also called `InteractionHandlingOutboundStarted`, Stat Server generates this momentary action when an agent accepts an outbound interaction. `HandlingOutboundStarted` is similar to `CallOutboundStarted` in the telephony model.

Back to top

HandlingStarted

This momentary action, also called `InteractionHandlingStarted`, marks the onset of interaction handling (`Handling`) for any interaction type, and it occurs when an agent accepts an inbound, outbound, or internal interaction. This action has no equivalent in the telephony model.

`HandlingStarted` is always simultaneous with one of the following interaction-type actions:

- `HandlingInboundStarted`
- `HandlingInternalStarted`
- `HandlingOutboundStarted`

The interaction type that Stat Server receives from Interaction Server with `EventPartyAdded` where `attr_party_type = 2` determines which one of the above three actions occurs simultaneously with `HandlingStarted`.

Back to top

InteractionAbandoned (Tenants)

This retrospective action is unconditionally generated on a Tenant upon receiving the `EventAbandoned`.

Mandatory event attribute `attr_actor_tenant_id` specifies the Tenant dbid.

Action duration is calculated from the moment when the interaction enters the system (specified by the `attr_itx_received_at` attribute) to the moment specified by the `attr_itx_abandoned_at` attribute. This attribute is initialized when the interaction is abandoned and appears in subsequent events.

As soon as Interaction Server recognizes an interaction as abandoned, it sends `EventAbandoned` to Stat Server, though the abandoned interaction might eventually be handled by an agent.

Interaction Server release 8.5.110.10 (or later) and Chat Media Server release 8.5.201.07 (or later) are required. This action was introduced in release 8.5.108.

Back to top

InteractionAbandonedDuringOffering (MediaChannels)

This retrospective action is unconditionally generated on agent (or place) upon receiving from Interaction Server the `EventRejected` event if the interaction is not accepted by an agent for any reason while offering or the `EventRevoked` event if the interaction is revoked.

Action duration is an interval between the `EventAgentInvited` and `EventRejected`/`EventRevoked` events. This action was introduced in release 8.5.102.

Back to top

InteractionAbandonedDuringOffering (StagingAreas)

This retrospective action is unconditionally generated on a StagingArea (Interaction Queue) upon receiving the `EventRevoked` event if the interaction is revoked or upon receiving the `EventRejected` event if the interaction is not accepted by an agent for any reason.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

Action duration is calculated from the time specified in the `attr_itx_placed_in_queue_at` attribute of the `EventRevoked` or `EventRejected` event to the time when these events occur. This action was introduced in release 8.5.101

Important

Starting with release 8.5.102, introduction of the `InteractionAbandonedDuringOffering` media-action on Agent/Pplace, StagingArea, VirtualQueue and Tenant adds more flexibility on counting all interactions that have been dropped (rejected or revoked) before they were accepted. Filters by reason can be applied to accommodate different conditions.

.

Back to top

InteractionAbandonedDuringOffering (Tenants)

This retrospective action is unconditionally generated on a Tenant upon receiving the `EventRevoked` event if the interaction is revoked or upon receiving the `EventRejected` event if the interaction is not accepted by an agent for any reason.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant dbid.

Action duration is calculated from the moment when the interaction enters the system (specified by the `attr_itx_received_at` attribute) to the moment when the interaction is revoked or rejected. This action was introduced in release 8.5.102.

Back to top

InteractionAbandonedDuringOffering (Virtual Queues)

For `Virtual Queue` objects, that are controlled by a Multimedia-monitored switch, the retrospective action `InteractionAbandonedDuringOffering` is unconditionally generated on an associated virtual queue upon receiving the `EventRevoked` event if the interaction is revoked or upon receiving the `EventRejected` event if the interaction is not accepted by an agent for any reason.

A virtual queue, associated with the interaction, is specified by the tracking details initially stored upon receiving the `EventCustom` event with packed `EventQueued` TEvent.

Action duration is calculated from the moment when the interaction has entered a virtual queue to the moment when the interaction is revoked or rejected. This action was introduced in release 8.5.102

Tip

To satisfy the RONA (Revoke On No Answer) requirement a filter by the `Expired` reason has to be used.

.

Back to top

InteractionAbandonedDuringOffering (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the EventRevoked or EventRejected event from Interaction Server.

Interaction attributes attr_itx_workbin_type_id and attr_itx_agent_id specify the Agent Workbin object.

Action duration is calculated from the time specified in the attr_itx_placed_in_queue_at attribute of the EventRevoked or EventRejected event to the time when these events occur. This action was introduced in release 8.5.112.10.

Back to top

InteractionAccepted1

This retrospective action is generated on an Agent and a Place independently upon receiving the EventPartyAdded event when the interaction is accepted for the very first time. Therefore, this action is generated only when the mandatory attribute attr_itx_delivered_at is equal to NULL.

Action duration is calculated from the moment when the interaction enters the system (specified by the attr_itx_received_at attribute) to the moment when the interaction is accepted for handling.

To apply this action to the Tenant, a propagation mechanism is used. A request from the client has to include a name of the Tenant object, but the object type needs to be GroupAgents or GroupPlaces.

> **Tip**
>
> The InteractionAccepted1 action is different from the Accepted (InteractionAccepted) action, which is also generated on agent/place objects.

This action was introduced in release 8.5.102.

Back to top

InteractionAccepted (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the EventPartyAdded event when an interaction is accepted by an agent for the first time. It is generated only ones for any interaction.

Mandatory event attribute attr_itx_queue specifies the Interaction Queue.

Interaction related attribute attr_itx_delivered_at provides the time stamp when an interaction is accepted for the first time.

The InteractionAccepted action is generated only if the attr_itx_delivered_at attribute is equal to NULL. **Note:** The attr_itx_delivered_at attribute provides the time stamp when an interaction was accepted for the first time.

Action duration is calculated from the moment when the interaction enters the system (specified by the `attr_itx_received_at` attribute) to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.101.

Back to top

InteractionAccepted (Tenants)

This retrospective action is generated on a Tenant upon receiving the `EventPartyAdded` event when an interaction is accepted by an agent for the first time. It is generated only ones for any interaction.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant's dbid.

Interaction related attribute `attr_itx_delivered_at` provides the time stamp when an interaction is accepted for the first time.

The `InteractionAccepted` action is generated only if the `attr_itx_delivered_at` attribute is equal to `NULL`. **Note:** The `attr_itx_delivered_at` attribute provides the time stamp when an interaction was accepted for the first time.

Action duration is calculated from the moment when the interaction enters the system (specified by the `attr_itx_received_at` attribute) to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.103.

Back to top

InteractionAccepted (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the `EventPartyAdded` event when an interaction is accepted by an agent for the first time. It is generated only once for any interaction based on the attribute `attr_itx_delivered_at` equal to NULL.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Action duration is calculated from the moment when the interaction enters the system (specified by the `attr_itx_received_at` attribute) to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.112.10.

Back to top

InteractionAgentPartyInProgress (Tenants)

This durable action is started on a Tenant when an interaction is initially associated with an agent(s). This action ends when there are no agents associated with the interaction.

The `InteractionAgentPartyInProgress` action can be started and stopped several times for the same interaction, but at any particular time there can be only one `InteractionAgentPartyInProgress` action with the specific `InteractionID`.

The `InteractionAgentPartyInProgress` action starts when Stat Server receives one of the following Interaction Server events:

- EventAgentInvited
- EventPartyAdded

Previously started `InteractionAgentPartyInProgress` action ends when Stat Server receives one of the following Interaction Server events:

- EventRejected
- EventRevoked
- EventPartyRemoved
- EventProcessingStopped

The following options are available in Stat Server to control memory usage, associated with the `InteractionAgentPartyInProgress` action:

- **[statserver]**/interaction-agent-party-in-progress-on-tenant-max-number
- **[statserver]**/interaction-agent-party-in-progress-on-tenant-media-list

This action was introduced in release 8.5.110.18.

Back to top

InteractionAnswered (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the `EventPartyAdded` event when an interaction is accepted by an agent.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

Action duration is calculated from the moment when the interaction is placed in the Interaction Queue (specified by the `attr_itx_placed_in_queue_at` attribute) to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.101.

Back to top

InteractionAnswered (Tenants)

This retrospective action is generated on a Tenant upon receiving the `EventPartyAdded` event when an interaction is accepted by an agent.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant's dbid.

Action duration is calculated from the moment when the interaction is placed in the Interaction Queue (specified by the `attr_itx_placed_in_queue_at` attribute) to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.103.

Back to top

InteractionAnswered (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the `EventPartyAdded` event when an interaction is accepted by an agent.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Action duration is calculated from the moment when the interaction is placed in the Agent Workbin to the moment when the interaction is accepted by an agent for handling. This action was introduced in release 8.5.112.10.

Back to top

InteractionCleared

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the `EventTakenFromQueue` event, when the interaction is diverted from the specific Interaction Queue to be processed not by an agent based on Actor information (`attr_party_type != 2`) provided in the event. **Note:** At this point an association between the Interaction Queue and the interaction still exists.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

Action duration is calculated from the moment when the interaction is placed in the Interaction Queue (specified by the `attr_itx_placed_in_queue_at` attribute) to the moment when the interaction leaves the Interaction Queue. This action was introduced in release 8.5.1.

Back to top

InteractionCleared (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the `EventTakenFromWorkbin` event, when the interaction is diverted not by an agent as specified by event attribute `attr_party_type != 2`.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Action duration is calculated from the moment when the interaction is placed into the Agent Workbin to the moment when the interaction leaves it. This action was introduced in release 8.5.112.10.

Back to top

InteractionCreated (StagingAreas)

This momentary action is generated on a StagingArea (Interaction Queue) upon receiving the `EventInteractionSubmitted` event, when a new interaction is submitted to the Interaction Queue.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

> **Important**
>
> Stat Server option `subscribe-for-all-ixn-server-events` must be set to `true` for this event to be sent by Interaction Server.

This action was introduced in release 8.5.100.

Back to top

InteractionCreated (Tenants)

This momentary action is generated on a Tenant upon receiving the `EventInteractionSubmitted` event, when a new interaction is received by the system.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant's dbid.

> **Important**
>
> Stat Server option `subscribe-for-all-ixn-server-events` must be set to `true` for this event to be sent by Interaction Server.

This action was introduced in release 8.5.102.

Back to top

InteractionDeleted (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the `EventProcessingStopped` event, when the interaction is finished and no longer exists in the system.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

Action duration is calculated from the moment of accepting the interaction by the system (specified by the `attr_itx_received_at` event attribute) to the moment when the interaction handling is finished by the system. This action was introduced in release 8.5.100.

Back to top

InteractionDeleted (Tenants)

This retrospective action is generated on a Tenant upon receiving the `EventProcessingStopped` event, when the interaction is finished and no longer exists in the system.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant's dbid.

Action duration is calculated from the moment when the interaction is accepted by the system (specified by the `attr_itx_received_at` event attribute) to the moment when the handling of this interaction is finished by the system. This action was introduced in release 8.5.102.

Back to top

InteractionDistributed

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the `EventTakenFromQueue` event, when the interaction is diverted from the specific Interaction Queue to be processed by an agent based on Actor information (`attr_party_type = 2`) provided in the event. **Note:** At this point an association between the Interaction Queue and the interaction still exists.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

Action duration is calculated from the moment when the interaction is placed in the Interaction Queue (specified by the `attr_itx_placed_in_queue_at` event attribute) to the moment when the interaction leaves the Interaction Queue. This action was introduced in release 8.5.1.

Back to top

InteractionDistributed (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the `EventTakenFromWorkbin` event, when the interaction is diverted by an agent as specified by the event attribute `attr_party_type = 2`.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Action duration is calculated from the moment when the interaction is placed into the Agent Workbin to the moment when the interaction leaves it. This action was introduced in release 8.5.112.10.

Back to top

InteractionDistributedToQueue

This retrospective action is generated on a StagingArea (Interaction Queue1 specified with the `attr_old_queue` attribute) upon receiving the `EventPlacedInQueue` event on an Interaction Queue2 (specified with the `attr_itx_queue` attribute) when the interaction is entered into the Interaction Queue2 from the Interaction Queue1.

The mandatory `attr_old_queue` event attribute specifies the Interaction Queue from where the interaction has been moved and must be different from the mandatory `attr_itx_queue` event attribute.

Action duration is calculated from the moment when the interaction was placed in an Interaction Queue1 (specified by the attribute `attr_itx_placed_in_queue_at` of the `EventPlacedInQueue` event on an Interaction Queue2) to the moment when the interaction was placed in the Interaction Queue2.

> **Important**
>
> Interaction Server v. 8.5.105.00 or later is required for calculation of the `InteractionDistributedToQueue` action duration.

This action was introduced in release 8.5.1.

Back to top

InteractionEntered

This momentary action is generated on a StagingArea (Interaction Queue) upon receiving the `EventPlacedInQueue` event or the `EventSubmitted` event with `attr_itx_state = 0`, when the interaction is entered into the specific Interaction Queue.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue. This action was introduced in release 8.5.1.

Back to top

InteractionEntered (Agent Workbin)

This momentary action is generated on an Agent Workbin upon receiving the `EventPlacedInWorkbin` event from Interaction Server.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object. This action was introduced in release 8.5.112.10.

Back to top

InteractionPastAcceptServiceLevel (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the `EventProcessingStopped` event with `UserData` containing the `ServiceObjective` key. A value of the `ServiceObjective` key provides expected duration for an interaction to be accepted for handling.

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

The `InteractionPastAcceptServiceLevel` action is generated in the following cases:

- For answered interactions - when the time interval `t1` between the moment of when the interaction is entering the system (specified by the `attr_itx_received_at` attribute) to the moment of the interaction acceptance by an agent (specified by the `attr_itx_delivered_at` attribute) is greater than the value of the `ServiceObjective` key.
- For the interactions without acceptance, such as revoked by customer (the `attr_itx_delivered_at` attribute is not available) - when the time interval `t2` between the moment of when the interaction is entering the system (specified by the `attr_itx_received_at` attribute) to the moment when the interaction is stopped is greater than the value of the `ServiceObjective` key.

For StagingArea objects, Stat Server generates this action only for the last StagingArea (Interaction Queue) that is associated with the interaction.

The actual duration of the InteractionPastAcceptServiceLevel action is calculated as a difference between t1 or t2 and the expected duration, specified by the ServiceObjective key. This action was introduced in release 8.5.102.

Back to top

InteractionPastAcceptServiceLevel (Tenants)

This retrospective action is generated on a Tenant upon receiving the EventProcessingStopped event with UserData containing the ServiceObjective key. A value of the ServiceObjective key provides expected duration for an interaction to be accepted for handling.

Mandatory event attribute attr_itx_tenant_id specifies the Tenant's dbid.

The InteractionPastAcceptServiceLevel action is generated in the following cases:

- For answered interactions - when the time interval t1 between the moment of when the interaction is entering the system (specified by the attr_itx_received_at attribute) to the moment of the interaction acceptance by an agent (specified by the attr_itx_delivered_at attribute) is greater than the value of the ServiceObjective key.
- For the interactions without acceptance, such as revoked by customer (the attr_itx_delivered_at attribute is not available) - when the time interval t2 between the moment of when the interaction is entering the system (specified by the attr_itx_received_at attribute) to the moment when the interaction is stopped is greater than the value of the ServiceObjective key.

The actual duration of the InteractionPastAcceptServiceLevel action is calculated as a difference between t1 or t2 and the expected duration, specified by the ServiceObjective key. This action was introduced in release 8.5.102.

Back to top

InteractionPastCompletionServiceLevel (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the EventProcessingStopped event with UserData containing the CompleteServiceObjective key. A value of the CompleteServiceObjective key provides expected duration to complete interaction handling.

Mandatory event attribute attr_itx_queue specifies the Interaction Queue.

The t1 time interval is the interval from the moment when the interaction is accepted by the system (specified by the attr_itx_received_at attribute) to the moment when the interaction handling is finished by the system.

The InteractionPastCompletionServiceLevel action is generated only if t1 is greater than the value of the CompleteServiceObjective key.

For StagingArea objects, Stat Server generates this action only for the last StagingArea (Interaction Queue) that is associated with the interaction.

The action duration is calculated as a difference between the t1 time interval and the expected duration, specified by the CompleteServiceObjective key. This action was introduced in release 8.5.102.

Back to top

InteractionPastCompletionServiceLevel (Tenants)

This retrospective action is generated on a Tenant upon receiving the EventProcessingStopped event with UserData containing the CompleteServiceObjective key. A value of the CompleteServiceObjective key provides expected duration to complete interaction handling.

Mandatory event attribute attr_itx_tenant_id specifies the Tenant's dbid.

The t1 time interval is the interval from the moment when the interaction is accepted by the system (specified by the attr_itx_received_at attribute) to the moment when the interaction handling is finished by the system.

The InteractionPastCompletionServiceLevel action is generated only if t1 is greater than the value of the CompleteServiceObjective key.

The action duration is calculated as a difference between the t1 time interval and the expected duration, specified by the CompleteServiceObjective key. This action was introduced in release 8.5.102.

Back to top

InteractionReleased (StagingAreas)

This retrospective action is generated on a StagingArea (Interaction Queue) upon receiving the EventPartyRemoved event.

Mandatory event attribute attr_old_queue specifies the Interaction Queue.

Action duration is calculated from the moment when the interaction is accepted by an agent (specified by the attr_itx_assigned_at attribute) to the moment when this agent no longer handles the interaction (**Note:** The interaction can still be alive).

> **Important**
>
> Interaction Server v. 8.5.105.00 or later is required for calculation of the InteractionReleased action duration.

This action was introduced in release 8.5.101.

Back to top

InteractionReleased (Tenants)

This retrospective action is generated on a Tenant upon receiving the `EventPartyRemoved` event.

Mandatory event attribute `attr_itx_tenant_id` specifies the Tenant's dbid.

Action duration is calculated from the moment when the interaction is accepted by an agent (specified by the `attr_itx_assigned_at` attribute) to the moment when this agent no longer handles the interaction (**Note:** The interaction can still be alive).

> ### Important
> Interaction Server v. 8.5.105.00 or later is required for calculation of the `InteractionReleased` action duration.

This action was introduced in release 8.5.103.

Back to top

InteractionReleased (Agent Workbin)

This retrospective action is generated on an Agent Workbin upon receiving the `EventPartyRemoved` event from Interaction Server.

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Action duration is calculated from the moment when the interaction is accepted by an agent (specified by the `attr_itx_assigned_at` attribute) to the moment when this agent no longer handles the interaction (the interaction can still exist). This action was introduced in release 8.5.112.10.

Back to top

InteractionResponded (Regular DNs)

This instant action is generated on a regular DN upon receiving the `EventReleased` TEvent with the following event attributes:

- `CallType != Outbound`
- `ThisDNRole = Destination` or `ThisDNRole = ConferenceMember`
- `CallState = Ok` or `CallState = Transferred`.

This action was introduced in release 8.5.105.

Back to top

InteractionResponded (MediaChannels)

This instant action is generated on an agent (or place) upon receiving the `EventPartyRemoved` event for:

- Any interaction of the `Inbound` or `Internal` type
- Any interaction of the `Outbound` type with the `attr_itx_subtype = OutboundReply`

> **Tip**
>
> This action is applicable to any agent involved in an interaction of the `Internal` type.

This action was introduced in release 8.5.105.

Back to top

InteractionWait (StagingAreas)

The `InteractionWait (StagingAreas)` durable action starts on a StagingArea when Stat Server receives the following events:

- `EventPlacedInQueue`
- `EventSubmitted` with `attr_itx_state=0`

The `InteractionWait (StagingAreas)` action ends upon the following events:

- `EventTakenFromQueue`
- `EventProcessingStopped`

Mandatory event attribute `attr_itx_queue` specifies the Interaction Queue.

The `interaction-wait-on-sa-max-number` configuration option controls the number of `InteractionWait (StagingAreas)` actions. When the threshold is reached, Stat Server does not create any more instances of the `InteractionWait (StagingAreas)` action, until the number of `InteractionWait (StagingAreas)` actions is below the threshold.

Upon a disconnect from Interaction Server all active `InteractionWait (StagingAreas)` actions across all StagingArea(s) end, which is reflected in the Stat Server log by a single message. This action was introduced in release 8.5.103.

Back to top

InteractionWait (Tenants)

The `InteractionWait (Tenants)` action is tracking an interaction until it is handled by an agent. This action can start and end multiple times for the same interaction.

This durable action starts when Stat Server receives one of the following Interaction Server events:

- EventInteractionSubmitted
- EventPlaceAgentState for a delivering interaction only with the `'Inviting' [int] = 1` attribute
- EventAgentInvited
- EventPartyRemoved with the `attr_operation` not equal to STOP (value 10), `attr_actor_type=AGENT` (value 2), and an agent is the last participant who is leaving the interaction

> ### Tip
> The value of the `attr_operation` attribute, different from STOP, means that an interaction is not finished and intended to be processed further by the system.

Previously started `InteractionWait (Tenants)` action ends when Stat Server receives one of the following Interaction Server events:

- EventProcessingStopped
- EventPartyAdded with the `attr_actor_type=AGENT` (value 2)

Mandatory event attribute `attr_actor_tenant_id` specifies the Tenant.

Upon disconnect from Interaction Server all active `InteractionWait` actions across all monitored Tenants are ended and a single message is printed in the Stat Server log. This action was introduced in release 8.5.107.

Back to top

InteractionWait (Agent Workbin)

The `InteractionWait (Agent Workbin)` durable action starts on an Agent Workbin when Stat Server receives the `EventPlacedInWorkbin` event from Interaction Server.

The `InteractionWait (Agent Workbin)` action ends upon the following events:

- `EventTakenFromWorkbin`
- `EventProcessingStopped`

Interaction attributes `attr_itx_workbin_type_id` and `attr_itx_agent_id` specify the Agent Workbin object.

Upon a disconnect from Interaction Server all active `InteractionWait (Agent Workbin)` actions across all Agent Workbins end, which is reflected in the Stat Server log by a single message. This action was introduced in release 8.5.112.10.

Back to top

LoggedIn

This durable action starts when Stat Server detects agent login on a DN:

- The `EventAgentLogin` TEvent is received on a DN.
- Either the `EventRegistered` or `EventQueryAddress` TEvent is received on a DN for which the `Extensions` attribute contains the pair, `("AgentStatus", value)`, where `value` is greater than zero (0 signifies LoggedOut).

This action ends with `EventAgentLogout` or when the NotMonitored action starts.

Back to top

LoggedOut

This durable action starts with `EventAgentLogout` and ends either with `EventAgentLogin` or when the NotMonitored action starts. For multimedia DNs, this action is classified as media-independent.

> **Tip**
>
> See also DN Actions at Newly Registered DNs.

Back to top

Monitored (Regular DNs)

This durable action starts whenever `NotMonitored` terminates—that is, when Stat Server is connected to T-Server or SIP Server and the link between T-Server (or SIP Server) and the switch is up. This action ends when the `NotMonitored` action starts.

Back to top

Monitored (Mediation DNs)

`Monitored (Mediation DNs)` is a durable, non–interaction-related actions that Stat Server generates to mediation DNs.

This action starts whenever `NotMonitored (Mediation DNs)` terminates—that is, when Stat Server is connected to T-Server (or SIP Server) and the link between the T-Server (or SIP Server) and the switch is up. This action ends when the `NotMonitored (Mediation DNs)` action starts.

For Stat Server 8.5.0 and higher releases (8.5.0$^+$), `Monitored (Mediation DNs)` durable action is generated to a Virtual Queue on the multimedia switch of the tenant if and only if there is at least one connected Interaction Server for that tenant. Otherwise, durable action NotMonitored (Mediation DNs) is generated.

Back to top

MonitoringInitiated

Stat Server generates this momentary action when an agent monitors an interaction.

Back to top

NotMonitored (Regular DNs)

This durable action begins whenever Stat Server is not connected to the T-Server or SIP Server controlling the switch where the DN is located (Stat Server receives the `EventServerDisconnected` TEvent in this case), or when the link between the T-Server (or SIP Server) and the switch is down (T-Server sends the `EventLinkDisconnected` TEvent). `NotMonitored` ends when both connections are up and running. Its complementary action is Monitored (Regular DNs) —one and only one of these actions can occur for any DN at any given moment. The `NotMonitored` action terminates every other DN action; no other DN action can start while `NotMonitored` is occurring.

Of special note, if Stat Server receives `EventOutOfService` for a particular DN (such as might be the case if the DN's switch is being reconfigured), the `NotMonitored` action occurs, and it persists until Stat Server detects `EventBackInService` for that DN. At that point, the `NotMonitored` action ceases.

Back to top

NotMonitored (Mediation DNs)

For Stat Server 8.1.2 and lower releases (8.1.2^{-}), this durable action begins whenever Stat Server is not connected to the T-Server controlling the switch where the DN is located (Stat Server receives the `EventServerDisconnected` TEvent in this case) or whenever the link between the T-Server and the switch is down (`EventLinkDisconnected` is received from T-Server). `NotMonitored (Mediation DNs)` ends when both connections are up.

Stat Server 8.5.0 and higher releases (8.5.0^{+}) generates this durable action on a Virtual Queue on the multimedia switch of the tenant if no Interaction Servers that are serving that tenant are connected to the Stat Server.

Its complementary action is Monitored (Mediation DNs). One and only one of these actions occurs for any DN at any moment. `NotMonitored (Mediation DNs)` terminates every other DN action; no other action can start while `NotMonitored (Mediation DNs)` is occurring.

Back to top

NotReadyForNextCall

This durable action is complementary to `WaitForNextCall` while the Monitored action occurs at the DN in question. Thus, `NotReadyForNextCall` occurs when `Monitored` occurs and one of the following conditions is met:

- Stat Server receives `EventRegistered` or `EventAddressInfo` with reports of agent status equal to either of the following:
    - `1 (LOGGED_IN)`

  - 3 (NOT_READY)
- Stat Server receives EventAgentLogin.
- Stat Server receives EventAgentNotReady with Workmode!=ACW while the agent is logged in.
- Stat Server receives EventDNDon.

The NotReadyForNextCall action ends when any of the following occur:

- Stat Server receives EventAgentReady (the WaitForNextCall action begins).
- Stat Server receives EventAgentNotReady with WorkMode=ACW (after-call work begins).
- Stat Server receives EventDNDoff while the agent is logged out, ready, or not ready with Workmode=ACW.
- The NotMonitored action starts.

The UserData, Reasons, and Extensions attributes from the EventDNDOn or EventDNDOff TEvents are not inherited by this action.

For multimedia DNs, this action is classified as media-dependent, media-unique.

> **Important**
>
> Agents cannot selectively make some media channels of a DN ready or not ready. These states apply to all of a DN’s media channels. For multimedia DNs, when conditions are met, Stat Server globally generates or ends the NotReadyForNextCall action for all enabled media channels supported by that DN.

Back to top

NotRoutable

This durable action is generated on an agent or place for a particular MediaType if the agent/place has DNs/media channels for this MediaType and the agent/place capacity does not allow routing to this agent/place for this MediaType. As soon as one of these conditions is not true, the NotRoutable action stops.

The NotRoutable action is generated on agent/place itself (not inherited from underlying DNs/media-channels) and then is propagated from an agent to the agent group or from a place to the place group.

The NotRoutable action is always associated with a MediaType and one NotRoutable action for a particular MediaType may exist on an agent/place. For a particular MediaType, the NotRoutable action is mutually exclusive with the Routable action.

Only NotRoutable and Routable actions support current_number, max_number, media_state and routable system attributes.

> **Important**
>
> - The NotRoutable action can only be used with Subject=DNAction.
> - The NotRoutable action does not affect the status of an agent or place.
> - To specify a MediaType Genesys recommends to use a filter with the corresponding system attribute, for example, PairExists(System, "MediaType", "chat").

This action was introduced in release 8.5.104.

Back to top

OffHook

This durable action starts when Stat Server receives EventOffHook from T-Server or SIP Server, and ends when Stat Server receives EventOnHook or the NotMonitored action starts. For DNs that generate these events, OnHook and OffHook are complementary while Monitored occurs. For multimedia DNs, this action is classified as media-independent.

> **Important**
>
> Stat Server ignores EventOffHook TEvent notifications if the ignore-off-hook-on-position Stat Server configuration option is set to true and the DN’s type is Position.

Back to top

OnHook

This durable action starts when Stat Server receives EventOnHook from T-Server or SIP Server, and ends when Stat Server receives EventOffHook or the NotMonitored action starts. This action is specific to a limited number of switches, and only DNs corresponding to physical telephones should be set to generate the corresponding TEvents. For such DNs, OnHook and OffHook are complementary while Monitored occurs. For multimedia DNs, this action is classified as media-independent.

> **Important**
>
> Stat Server ignores EventOnHook TEvent notifications if the ignore-off-hook-on-position Stat Server configuration option is set to true and the DN’s type is Position.

Back to top

OrigDNCallAbandoned

This agent group and place group action occurs at the same time as a CallAbandoned action, which occurs at a mediation DN configured as an origination DN for the group. `OrigDNCallAbandoned` relates to the same interaction as the corresponding `CallAbandoned` action.

`OrigDNCallAbandoned` is a retrospective group action reflecting origination DNs that Stat Server generates to agent and place groups.

`OrigDNCallAbandoned` is always simultaneous with one of the following calltype actions:

- `OrigDNCallAbandonedUnknown`
- `OrigDNCallAbandonedInternal`
- `OrigDNCallAbandonedInbound`
- `OrigDNCallAbandonedOutbound`
- `OrigDNCallAbandonedConsult`

The interaction type that Stat Server receives from T-Server with `EventAbandoned` determines which of the above five actions occurs simultaneously with `OrigDN CallAbandoned`.

Back to top

OrigDNCallDistributed

This agent group and place group action occurs at the same time as a CallDistributed action, which occurs at a mediation DN configured as an origination DN for the group. `OrigDNCallDistributed` relates to the same interaction as the corresponding `CallDistributed` action.

`OrigDNCallDistributed` is a retrospective group action reflecting origination DNs that Stat Server generates to agent and place groups.

`OrigDNCallDistributed` is always simultaneous with one of the following call-type actions:

- `OrigDNCallDistributedUnknown`
- `OrigDNCallDistributedInternal`
- `OrigDNCallDistributedInbound`
- `OrigDNCallDistributedOutbound`
- `OrigDNCallDistributedConsult`

The interaction type that Stat Server receives from T-Server with Event `Diverted` or `EventRouteUsed` determines which of the above five actions occurs simultaneously with `OrigDNCallDistributed`.

Back to top

OrigDNCallEntered

This agent group and place group action occurs at the same time as a CallEntered action, which occurs at a mediation DN configured as an origination DN for the group. `OrigDNCallEntered` relates to the same interaction as the corresponding `CallEntered` action.

`OrigDNCallEntered` is a momentary group action reflecting origination DNs that Stat Server generates to agent and place groups.

`OrigDNCallEntered` is always simultaneous with one of the following call-type actions:

- `OrigDNCallEnteredUnknown`
- `OrigDNCallEnteredInternal`
- `OrigDNCallEnteredInbound`
- `OrigDNCallEnteredOutbound`
- `OrigDNCallEnteredConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with `OrigDNCallEntered`.

Back to top

OrigDNCallWait

This agent group and place group action starts and ends at the same time as a CallWait action, which starts and ends at a mediation DN, configured as an origination DN for the group. `OrigDNCallWait` relates to the same interaction as the corresponding `CallWait` action.

You can list origination DNs on the `Advanced` tab of the `Properties` dialog box of an `Agent Group` or `Place Group` object. If you list queues and routing points from which calls are delivered to a given `Group` object as origination DNs for that group, you can use events occurring at such DNs in agent group and place group statistics. For this purpose, Stat Server reflects some mediation DN actions as a special set of agent and place group actions.

`OrigDNCallWait` is a durable group action reflecting origination DNs that Stat Server generates to agent and place groups.

`OrigDNCallWait` is always simultaneous with one of the following call-type actions:

- `OrigDNCallWaitUnknown`
- `OrigDNCallWaitInternal`
- `OrigDNCallWaitInbound`
- `OrigDNCallWaitOutbound`
- `OrigDNCallWaitConsult`

The interaction type that Stat Server receives from T-Server with `EventQueued` or `EventRouteRequest` determines which of the above five actions occurs simultaneously with

OrigDNCallWait.

Back to top

Pulled

Stat Server generates this momentary action, also called InteractionPulled, every time it detects that an interaction has been pulled from the interaction queue and directed to be delivered to a resource.

Back to top

Rejected

This retrospective action, also called InteractionRejected, is generated on an agent (or place) upon receiving the EventRejected event from Interaction Server and indicates that an agent has rejected the delivered interaction. This action terminates Delivering actions, and it is similar to the CallAbandonedFromRinging action in the telephony model.

Action duration is an interval between EventAgentInvited and EventRejected.

> ### Tip
> The CallAbandonedFromRinging action is a legacy alias for the Rejected action.

Back to top

Revoked

This retrospective action, also called InteractionRevoked, indicates that the system has revoked the interaction at the agent's desktop. This action has no equivalent in the telephony model.

Back to top

Routable

This durable action is generated on an agent or place for a particular MediaType if the agent/place has DNs/media channels for this MediaType and the agent/place capacity allows routing to this agent/place for this MediaType. As soon as one of these conditions is not true, the Routable action stops.

The Routable action is generated on agent/place itself (not inherited from underlying DNs/media-channels) and then is propagated from an agent to the agent group or from a place to the place group.

The Routable action is always associated with a MediaType and one Routable action for a particular MediaType may exist on an agent/place. For a particular MediaType, the Routable action is mutually exclusive with the NotRoutable action.

Only `Routable` and `NotRoutable` actions support `current_number`, `max_number`, `media_state` and `routable` system attributes.

> **Important**
>
> - The `Routable` action can only be used with `Subject=DNAction`.
> - The `Routable` action does not affect the status of an agent or place.
> - To specify a `MediaType` Genesys recommends to use a filter with the corresponding system attribute, for example, `PairExists( System, "MediaType", "chat")`.

This action was introduced in release 8.5.104.

Back to top

StartedInternal

This momentary action, also called `InteractionStartedInternal`, indicates that an agent has initiated an internal interaction. This action has no equivalent in the telephony model.

Back to top

StartedOutbound

This momentary action, also called `InteractionStartedOutbound`, indicates that an agent has initiated an outbound interaction. This action has no equivalent in the telephony model.

> **Important**
>
> There is no such `StartedInbound` action that Stat Server generates.

Back to top

StoppedInbound

This retrospective action, also called `InteractionStoppedInbound`, indicates that an agent has terminated an inbound interaction. This action has no equivalent in the telephony model.

Back to top

StoppedInternal

This retrospective action, also called `InteractionStoppedInternal`, indicates that an agent has terminated an internal interaction. This action has no equivalent in the telephony model.

Back to top

StoppedOutbound

This retrospective action, also called InteractionStoppedOutbound, indicates that an agent has terminated an outbound interaction. This action has no equivalent in the telephony model.

> **Important**
>
> Actions InteractionStoppedInbound, InteractionStoppedInternal, and InteractionStoppedOutbound are not generated upon receiving the EventProcessingStopped event if before stopping the interaction an agent was logged out of the media (indicated by EventMediaRemoved event).

Back to top

StuckCallCleaned

This retrospective action occurs at a mediation DN and derives from the CallWait durable action if Stat Server terminates the CallWait action because Stat Server receives the EventAbandoned TEvent from T-Server with an AttributeReliability attribute not equal to TReliabilityOk.

StuckCallCleaned is always simultaneous with one of the following call-type actions:

- StuckCallCleanedUnknown
- StuckCallCleanedInternal
- StuckCallCleanedInbound
- StuckCallCleanedOutbound
- StuckCallCleanedConsult

The interaction type that Stat Server receives from T-Server with EventQueued or EventRouteRequest determines which of the above five actions occurs simultaneously with StuckCallCleaned.

Back to top

StuckCallCleanedWhileRinging (Regular DNs)

This retrospective action derives from the CallRinging durable action if Stat Server receives EventAbandoned with an AttributeReliability attribute not equal to TReliabilityOk for the DN. This action’s corresponding initial momentary action is CallRingingStarted.

StuckCallCleanedWhileRinging is always simultaneous with one of the following call-type actions:

- `StuckCallCleanedWhileRingingUnknown`
- `StuckCallCleanedWhileRingingInternal`
- `StuckCallCleanedWhileRingingInbound`
- `StuckCallCleanedWhileRingingOutbound`
- `StuckCallCleanedWhileRingingConsult`

The interaction type that Stat Server receives from T-Server with `EventAbandoned` (with `AttributeReliability!=TReliabilityOk`) determines which of the above five actions occurs simultaneously with `StuckCallCleanedWhileRinging`.

Back to top

StuckCallCleanedWhileRinging (Mediation DNs)

This retrospective action derives from the CallRinging durable action and occurs at a mediation DN when Stat Server receives `EventAbandoned` with an `AttributeReliability` attribute other than `TReliabilityOk` from a DN to which an interaction was distributed from the mediation DN. `StuckCallCleanedWhileRinging` receives as its duration the interval from the moment when the interaction enters the mediation DN (`EventQueued` or `EventRouteRequest`) to the moment when Stat Server receives the `EventAbandoned` TEvent (with `AttributeReliability!=TReliabilityOk`). This action’s corresponding initial momentary action is CallRingingStarted.

`StuckCallCleanedWhileRinging` is always simultaneous with one of the following call-type actions:

- `StuckCallCleanedWhileRingingUnknown`
- `StuckCallCleanedWhileRingingInternal`
- `StuckCallCleanedWhileRingingInbound`
- `StuckCallCleanedWhileRingingOutbound`
- `StuckCallCleanedWhileRingingConsult`

The interaction type that Stat Server receives from T-Server with `EventReleased` (with `AttributeReliability!=TReliabilityOk`) determines which of the above five actions occurs simultaneously with `CallRetrievedFromHold`.

Back to top

TransferMade

This momentary action, also called `InteractionTransferMade`, indicates that an agent has transferred the interaction to another agent directly; that is, the transfer does not occur through a mediation DN. This action is similar to `CallTransferMade` in the telephony model.

`TransferMade` is always simultaneous with one of the following interaction-type actions:

- `TransferMadeInbound`
- `TransferMadeInternal`

- TransferMadeOutbound

The interaction type that Stat Server receives from Interaction Server with EventEstablished determines which of the above three actions occurs simultaneously with TransferMade.

Back to top

TransferTaken

This momentary action, also called InteractionTransferTaken, indicates that an agent has received the transferred interaction. This action is similar to CallTransferTaken in the telephony model.

Back to top

TransferredFromHold

This retrospective action derives from the CallOnHold durable action if CallOnHold terminates because of EventReleased with an interaction state of Transferred.

TransferredFromHold is always simultaneous with one of the following call-type actions:

- TransferredFromHoldUnknown
- TransferredFromHoldInternal
- TransferredFromHoldInbound
- TransferredFromHoldOutbound
- TransferredFromHoldConsult

The interaction type that Stat Server receives from T-Server with EventReleased determines which of the above five actions occurs simultaneously with TransferredFromHold.

Back to top

UserEvent (Regular DNs)

The EventUserEvent TEvent triggers this momentary, instantaneous action.

Starting with Stat Server release 8.5.103, the UserEvent (Regular DNs) action inherits GlobalUserData, Reasons and Extensions key-value lists from the EventUserEvent TEvent.

Back to top

UserEvent (Mediation DNs)

The EventUserEvent TEvent triggers the UserEvent momentary, instantaneous action, which is not related to an interaction, but which, like interaction-related actions, carries data that accompanies the TEvent. This means you can use this action in defining filtered statistics and custom-formula statistics.

Starting with Stat Server release 8.5.103, the `UserEvent (Mediation DNs)` action inherits `GlobalUserData`, `Reasons` and `Extensions` key-value lists from the `EventUserEvent` TEvent. This action was introduced in release 8.5.0.

Back to top

UserEventReceived (Regular DNs)

The `EventUserEvent` TEvent triggers the `UserEventReceived (Regular DNs)` durable action.

Upon receiving the `EventUserEvent` TEvent on an object, specified by the `ThisDN` attribute, the following occurs:

- If there is an `UserEventReceived (Regular DNs)` action in progress, that action is ended.
- New action `UserEventReceived (Regular DNs)` is started, inheriting `GlobalUserData`, `Reasons` and `Extensions` key-value lists from the event.

> **Important**
>
> - The `EventUserEvent (Regular DNs)` durable action is not used in status calculations on a DN/Queue.
> - The only supported `Subject` for the `UserEventReceived (Regular DNs)` action is `DNAction`.
> - `UserData` key-value list cannot be used in filters/formulas, that are applied to the `UserEventReceived (Regular DNs)` action.

This action was introduced in release 8.5.103.

Back to top

UserEventReceived (Mediation DNs)

The `EventUserEvent` TEvent triggers the `UserEventReceived (Mediation DNs)` durable action.

Upon receiving the `EventUserEvent` TEvent on an object, specified by the `ThisDN` attribute, the following occurs:

- If there is an `UserEventReceived (Mediation DNs)` action in progress, that action is ended.
- New action `UserEventReceived (Mediation DNs)` is started, inheriting `GlobalUserData`, `Reasons` and `Extensions` key-value lists from the event.

> **Important**

> - The EventUserEvent (Mediation DNs) durable action is not used in status calculations on a DN/Queue.
> - The only supported Subject for the UserEventReceived (Mediation DNs) action is DNAction.
> - UserData key-value list cannot be used in filters/formulas, that are applied to the UserEventReceived (Mediation DNs) action.

This action was introduced in release 8.5.103.

Back to top

WaitForNextCall

This durable action occurs for a particular DN, regardless of media channel, if all of the following conditions are met:

- Monitored occurs.
- The last TEvent to arrive after any of the following TEvents is EventAgentReady:
    - EventAgentLogin
    - EventAgentNotReady
    - EventRegistered
    - EventAddressInfo reports agent status 2 (Ready)
- Either EventDNDOn is never received, or the last event from the pair EventDNDOn and EventDNDOff is EventDNDOff.

The only exceptions to this rule are the DNs of type Extension or Voice Treatment Port, for which the WaitForNextCall action starts as soon as the DN is registered.

> **Tip**
>
> See also DN Actions at Newly Registered DNs.

WaitForNextCall ends on a DN when any of the following occurs:

- Stat Server receives EventRegistered or EventAddressInfo with reports of agent status equal to any of the following:
    - 0 (LoggedOut)
    - 3 (NOT_READY)
    - 4 (ACW)

    - 5 (Walk_Away)
- Stat Server receives EventDNDon.
- Stat Server receives EventDNOutOfService.
- Stat Server receives EventAgentNotReady with any work mode.
- Stat Server receives EventAgentLogout.
- The NotMonitored action starts.

While Monitored occurs, the actions WaitForNextCall, NotReadyForNextCall, and AfterCallWork are complementary.

The UserData, Reasons, and Extensions attributes from the EventDNDOn or EventDNDOff TEvents are not inherited by this action.

For multimedia DNs, this action is classified as media-dependent, media-unique.

> **Important**
>
> Agents cannot selectively make some media channels of a DN ready or not ready. These states apply to all of a DN’s media channels.

Back to top