Agent status and agent events

Log on/Log off/Pause/Return (ContactCentreStatus)

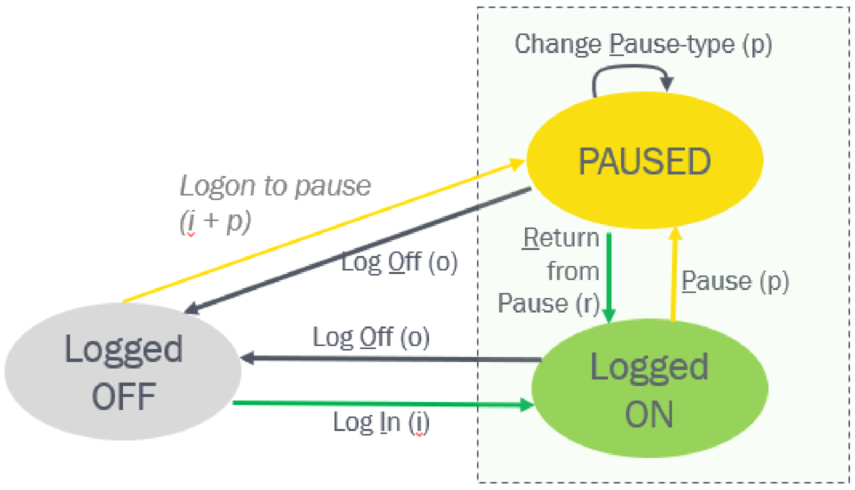
We do not create a raw data record when a user is signing in to Puzzel. Such sign-ins are shown in the Admin Portal's Access log.

The Puzzel agent (when signed in) can log on to queue, logon to pause, go to pause (from logged on), return from pause and log off. There are 3 different statuses:

- Logged on (and not paused)
- (Logged on and) Paused
- Logged off.

> Note
>
> In the Puzzel user interfaces the term 'Logged on' sometimes includes agents in Pause.

For each time an agent logs on/logs off/starts pause/returns from pause, an Agent event with the correct timestamp and event_type (i/o/p/r) is created. These agent events do not have any duration, so one must calculate to find the time spent in each status.



Web services - **ContactCentreStatus**: LoggedOff, LoggedOn, Pause

Pause can be used for any reason for not being ready. It is possible to define different Pause types like Lunch, Meeting, Admin etc, and if this is defined, the actual Pause type is included in the agent event for entering Pause. An agent can change status from one pause type to another pause type.

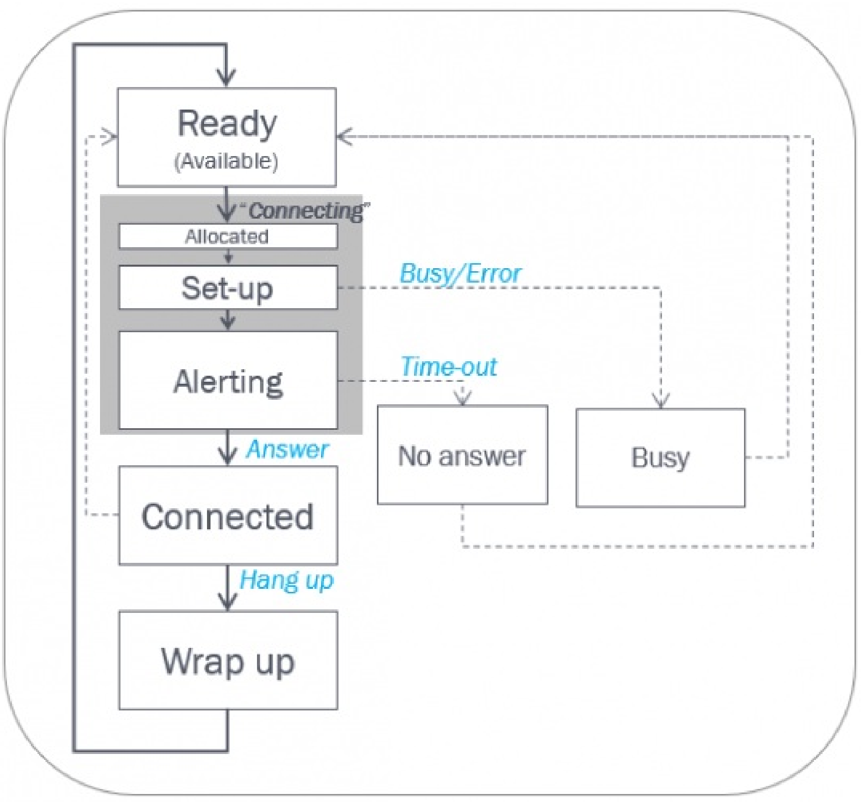
The agent logs on to queue with a profile. The agent may have different profiles, each representing one or more queues that the agent can answer. The agent's profile name, phone number and user group are shown in the agent event for logons (event_type=i), but not the actual queues the profile represents.

Please note that if you have a pause type called Wrap-up or Extra wrap-up (not recommended!), the time in this pause type is reported as time in status Pause, and not as time in status Wrap-up, so it will not be included in the reported Wrap-up and AHT in Puzzel real-time views and Puzzel statistics!

In agent.puzzel.com it is possible for agents to log on directly to pause (from January 2019). To avoid challenges with this new status transition (Off -> Paused) in any raw data queries or calculations, we decided to create both a login event and a pause event with dte_start some milliseconds apart when agent logs on directly to pause.

Agent status in the queue “engine”

When an agent logs on to the queue(s) she is set to status Ready (sometimes called Available). When a Puzzel queue allocates this agent and then calls (or sends a written request) to the agent, the agent’s status is first (Allocated and then) Set-up and then Alerting, but these statuses are presented as Connecting in the Puzzel user interface.



- If the agent answers, the agent status is set to *Connected*. At hang-up, the status changes to *Wrap up* (if > 0 sec wrap-up is configured) and then *Ready*.
- If the call to the agent's phone number results in busy (or error), the agent's status is set to *Busy* for the configured number of seconds (default 15) and then to *Ready*.
- If the call to an agent is not answered by the agent after x seconds ringing (default 30), the agent is set in status *No answer* for the configured number of seconds (default 15), and then back to *Ready*.

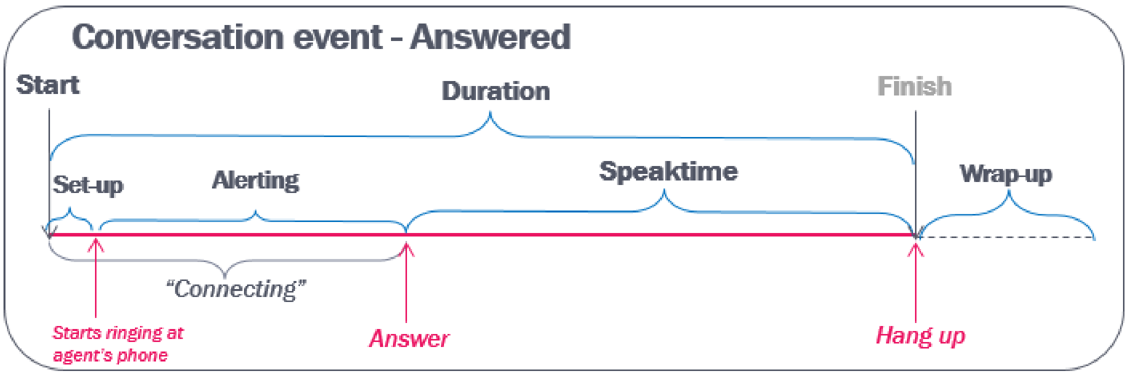
In the **Conversation** event we specify

- Start time (dte_start). At call set up (when written request was sent to agent)
- Duration (duration_tot_sec). From start until hang-up (until agent closes the written request tab),
- Speaktime (duration_speak_sec). From agent answers call (accepts the written request) until call hang-up (agent closes the written request tab),
- dte_speak_start (when speaktime starts)
- Wrap-up time (wrap_up_sec).
- And other details

Finish is shown in most illustrations and mentioned in the text, but Finish is not a field in the database. Finish is equal to Start (dte_start) + Duration (duration_tot_sec).

Note

The Wrap-up happens after the Conversation event’s Finish.



"Connecting" vs Set-up and Alerting

The connecting phase (=duration_tot_sec minus duration_speak_sec) consists of two phases; call set-up and alerting.

The call set-up phase is usually <0,5 sec when Puzzel softphone is used, 1-2 seconds when a landline phone is used, and maybe 3-8 seconds when calling to a mobile number.

The alerting phase ("ringing time") is usually very short (>0,7 sec) if the softphone auto-answers, and without auto-answer, the typical alerting time is 10-25 sec.

The lengths of the Set-up and Alerting phases are specified in the Conversation event (setup_ms and alert_ms).

Note

Some calls from the Puzzel platform do not receive the correct signalling events from other telecom operators or the local phone/SIP system, and this will affect the reported setup and alerting times. If Puzzel receives "answer" without a "ringing/alerting" signal first, this will usually result in a too high value for setup_ms (the reported set-up will be the sum of actual set-up and alerting time), and the value in alert_ms might be 0). If the call events from your Puzzel solution contains 'wrong' values for Set-up and Alerting, please use the connecting phase (=conversation event duration minus speaktime) as a proxy for ringing time when calculating ringing time at agent level.

Details about Wrap-up

Wrap-up is initially a pre-defined number of seconds, but the agent may shorten wrap-up by clicking Ready/Log off/Pause during wrap-up, or click to extend the wrap-up.

If the agent clicks *Ready* while in status Wrap-up, the initial reported Wrap-up time (wrap_up_sec) in the Conversation event will be corrected, normally after no more than 5 minutes (given that the whole session has ended).

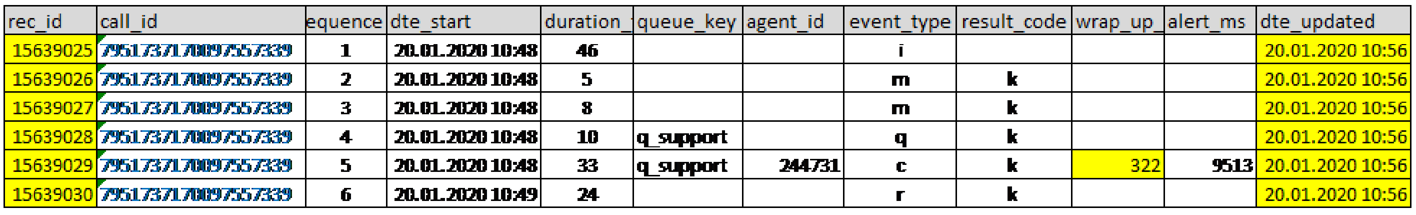
Example:

Shortly after call end, all records for the call_id including the conversation event with the long pre-defined wrap-up (e.g. 600 sec) can be found in table call_events.

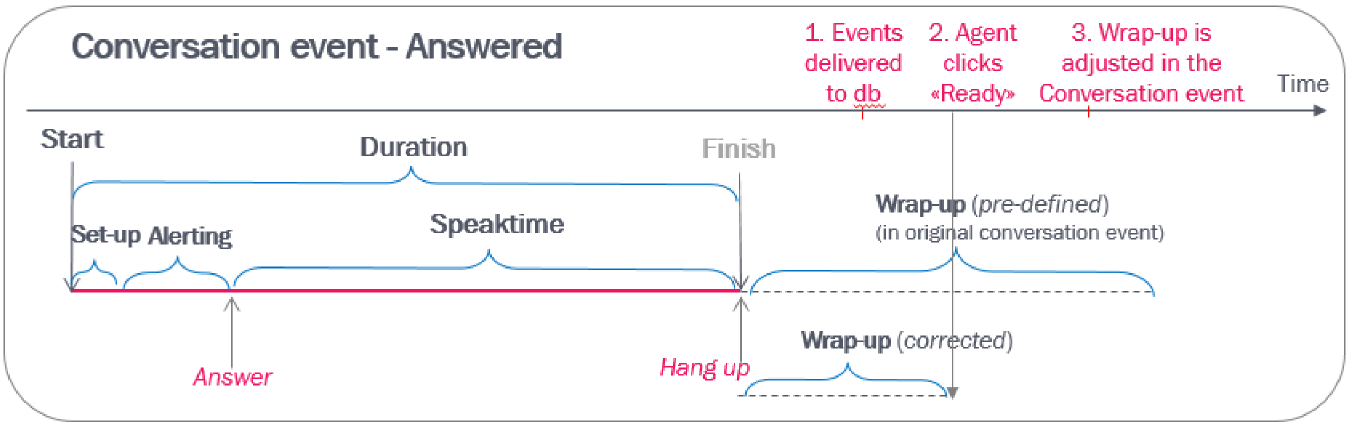
rec_id	call_id	equence	dte_start	duration_	queue_key	agent_id	event_type	result_code	wrap_up_	alert_ms	dte_updated
15638962	7951737170097557339	1	20.01.2020 10:48	46			i				20.01.2020 10:50
15638963	7951737170097557339	2	20.01.2020 10:48	5			m	k			20.01.2020 10:50
15638964	7951737170097557339	3	20.01.2020 10:48	8			m	k			20.01.2020 10:50
15638965	7951737170097557339	4	20.01.2020 10:48	10	q_support		q	k			20.01.2020 10:50
15638966	7951737170097557339	5	20.01.2020 10:48	33	q_support	244731	c	k	600	9513	20.01.2020 10:50
15638967	7951737170097557339	6	20.01.2020 10:49	24			r	k			20.01.2020 10:50

After a while, when the agent has (extended and/or) ended his wrap-up, all events for this call_id are **replaced** with new events with **new rec_ids** and a new dte_updated, and now the conversation event has the updated value for wrap-up (here

322 sec).



rec_id	call_id	equence	dte_start	duration_	queue_key	agent_id	event_type	result_code	wrap_up_	alert_ms	dte_updated
15639025	7951737170097557339	1	20.01.2020 10:48	46			i				20.01.2020 10:56
15639026	7951737170097557339	2	20.01.2020 10:48	5			m	k			20.01.2020 10:56
15639027	7951737170097557339	3	20.01.2020 10:48	8			m	k			20.01.2020 10:56
15639028	7951737170097557339	4	20.01.2020 10:48	10	q_support		q	k			20.01.2020 10:56
15639029	7951737170097557339	5	20.01.2020 10:48	33	q_support	244731	c	k	322	9513	20.01.2020 10:56
15639030	7951737170097557339	6	20.01.2020 10:49	24			r	k			20.01.2020 10:56


Conversation event - Answered
1. Events delivered to db
2. Agent clicks «Ready»
3. Wrap-up is adjusted in the Conversation event
Time
Start
Duration
Finish
Set-up
Alerting
Speaktime
Wrap-up (pre-defined)
(in original conversation event)
Answer
Hang up
Wrap-up (corrected)

If the agent can extend wrap, you will sometimes see that the initial reported wrap-up in a conversation event will be extended. One agent can click 'extend wrap-up' one or more times for one call.

If the agent clicks *Log off/Pause* while in wrap-up, the initial reported wrap-up in the Conversation event will be corrected, normally after no more than 5 minutes after the event happened (given that the session has ended).

Note

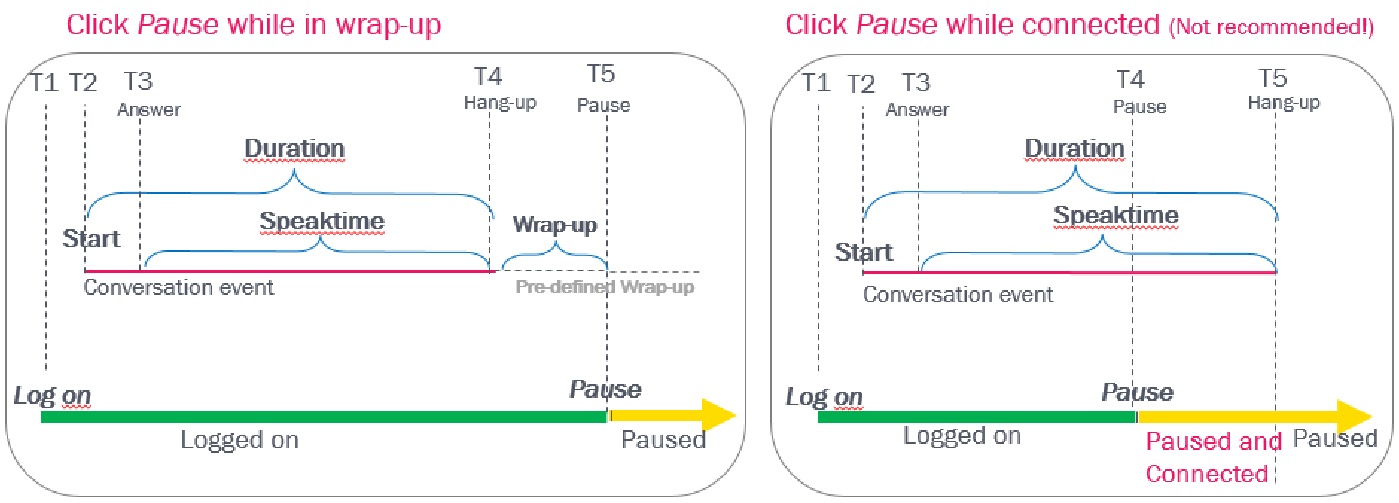
Conversation events with other results than answered (k) also have a value for wrap-up (the pre-defined value that would have been used if the agent answered the call), so when you calculate (average) wrap-up time for calls for agents, **make sure you only include wrap-up for answered conversation events** (result_code=k).

If you have a pause type called *Wrap-up* or *Extra wrap-up (Not recommended!)*, remember that the time in this pause type is reported as time in status Pause (event_type='p') , and not as time in status Wrap-up in conversation events!

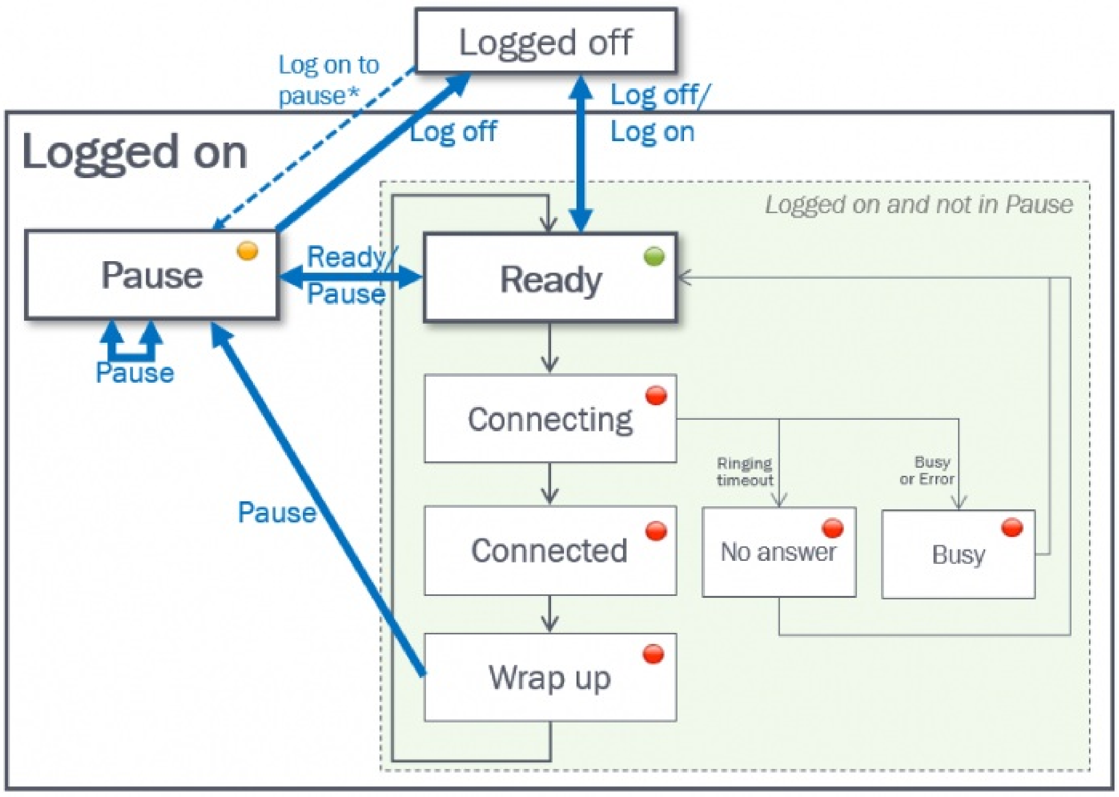
Connected while in status Pause or Logged off

If the agent clicks *Pause* (or Log off) while *Connected*, the agent will be in status [Paused AND Connected] at the same time (or [Logged off and Connected]). The reported wrap-up time will be 0, but in agent statistics the agent's time in pause will be "too" long.

The reason why an agent may click *Pause* while connected might be to avoid receiving a new call from queue just after having hung-up, since the plan is to have a pause just after call end. However, if the predefined wrap-up is > 15 seconds, agents should instead end the call and then click Pause while in status Wrap-up!



Normal status transitions:



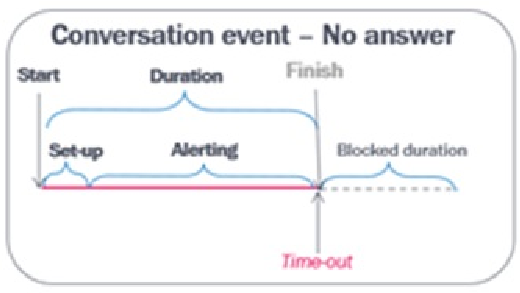
* If the agent clicks Logon with a pause type selected, the agent will be logged on and immediately put into pause (both logon and pause records are generated).

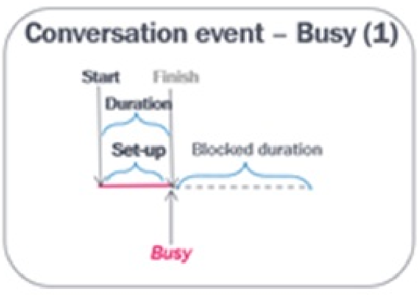
Busy and No-answer

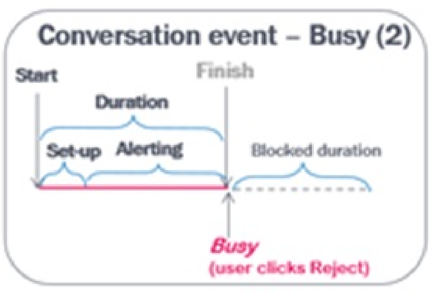
If a call (from a queue) to an agent results in busy or no-answer, the agent is set to status Busy/No-answer for a configurable number of seconds. This pre-defined Busy/No-answer timeout or ‘blocked’ time is from v.1.5.0.0 recorded in 'block_duration_sec' in the Conversation events for media_type_id=1 (phone) with result busy (b) and timeout (t).

Note

If the agent clicks Ready/Log off/Pause while in status Busy/No-answer, the agent status will change, but the block_duration_sec value will not be updated in the Conversation event (as we do with shortened/extended wrap-up).







'Audit log' – see all events for an agent (fnc_agent_events_window)

In table agent_events we have agent login (i)/logoff (o)/pause (p)/return from pause (r) events (with no duration) and conversation events with duration that includes information about connecting, connected/speaktime and wrap-up time.

If you want to see the "timeline" for one agent, you can use the function fnc_agent_events_window (seeFunctions and Stored Procedures) to list agent events for this agent, with start/duration and adjusted start/duration, chronologically.

Note

Conversation events for so-called 'secondary agents', that is, chat, social and email requests, are not included here!

If you use this function with **mode=1**;

- Login events are given adjusted duration until next log off (or period end)
- Logoff events are given adjusted duration until next logon (or period end if that happens first)
- Pause events are given adjusted duration until first Return from pause.
- Return from pause events are removed
- Answered (result k) and unanswered (result t, h, b…) Conversation events are listed
- Answered Conversation events with wrap-up time > 0 are divided into two:
    - one Conversation event
    - one Wrap-up event (event_type=w)*
- For the (ready) time between Conversation events we are generating Available events (event_type=a)*
- You can remove the Login events (which is implicit expressed by Available) from the result by adding "where event_type not in ('i')" to your query.

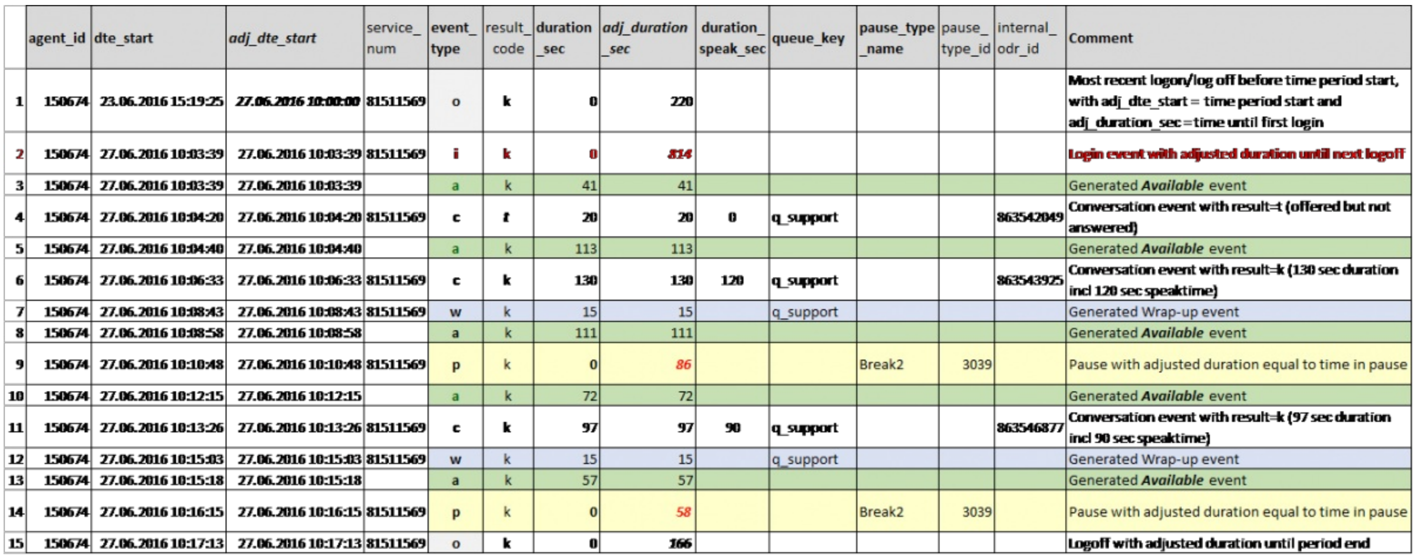
* Events with type Available (a) and Wrap-up (w) do not appear directly in the agent_events table.

Note

We do not create (blocked) events for the (usually few and short) time periods the agent is in status Busy/No-answer.

Example query and result:

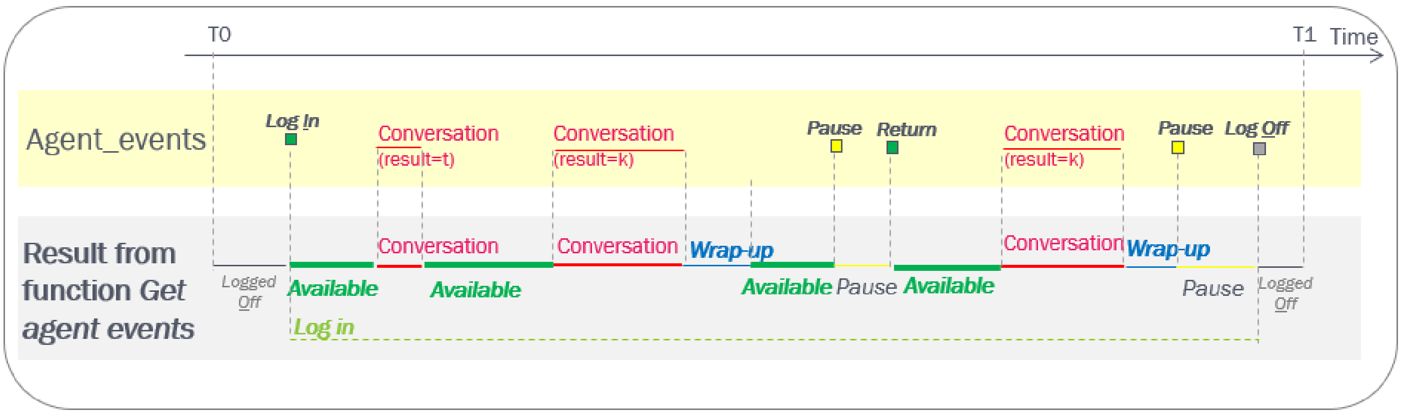
```
select * from [dbo].[fnc_agent_events_window]
(150674,'27-jun-2016 10:00','27-jun-2016 10:20',1, , )
order by dte_start
```



	agent_id	dte_start	*adj_dte_start*	service_num	event_type	result_code	duration_sec	*adj_duration_sec*	duration_speak_sec	queue_key	pause_type_name	pause_type_id	internal_odr_id	Comment
1	150674	23.06.2016 15:19:25	*27.06.2016 10:00:00*	81511569	o	k	0	220						Most recent logon/log off before time period start, with adj_dte_start = time period start and adj_duration_sec=time until first login
2	150674	27.06.2016 10:03:39	27.06.2016 10:03:39	81511569	i	k	0	*814*						Login event with adjusted duration until next logoff
3	150674	27.06.2016 10:03:39	27.06.2016 10:03:39		a	k	41	41						Generated *Available* event
4	150674	27.06.2016 10:04:20	27.06.2016 10:04:20	81511569	c	t	20	20	0	q_support			863542049	Conversation event with result=t (offered but not answered)
5	150674	27.06.2016 10:04:40	27.06.2016 10:04:40		a	k	113	113						Generated *Available* event
6	150674	27.06.2016 10:06:33	27.06.2016 10:06:33	81511569	c	k	130	130	120	q_support			863543925	Conversation event with result=k (130 sec duration incl 120 sec speaktime)
7	150674	27.06.2016 10:08:43	27.06.2016 10:08:43	81511569	w	k	15	15		q_support				Generated Wrap-up event
8	150674	27.06.2016 10:08:58	27.06.2016 10:08:58		a	k	111	111						Generated *Available* event
9	150674	27.06.2016 10:10:48	27.06.2016 10:10:48	81511569	p	k	0	*86*			Break2	3039		Pause with adjusted duration equal to time in pause
10	150674	27.06.2016 10:12:15	27.06.2016 10:12:15		a	k	72	72						Generated *Available* event
11	150674	27.06.2016 10:13:26	27.06.2016 10:13:26	81511569	c	k	97	97	90	q_support			863546877	Conversation event with result=k (97 sec duration incl 90 sec speaktime)
12	150674	27.06.2016 10:15:03	27.06.2016 10:15:03	81511569	w	k	15	15		q_support				Generated Wrap-up event
13	150674	27.06.2016 10:15:18	27.06.2016 10:15:18		a	k	57	57						Generated *Available* event
14	150674	27.06.2016 10:16:15	27.06.2016 10:16:15	81511569	p	k	0	*58*			Break2	3039		Pause with adjusted duration equal to time in pause
15	150674	27.06.2016 10:17:13	27.06.2016 10:17:13	81511569	o	k	0	*166*						Logoff with adjusted duration until period end

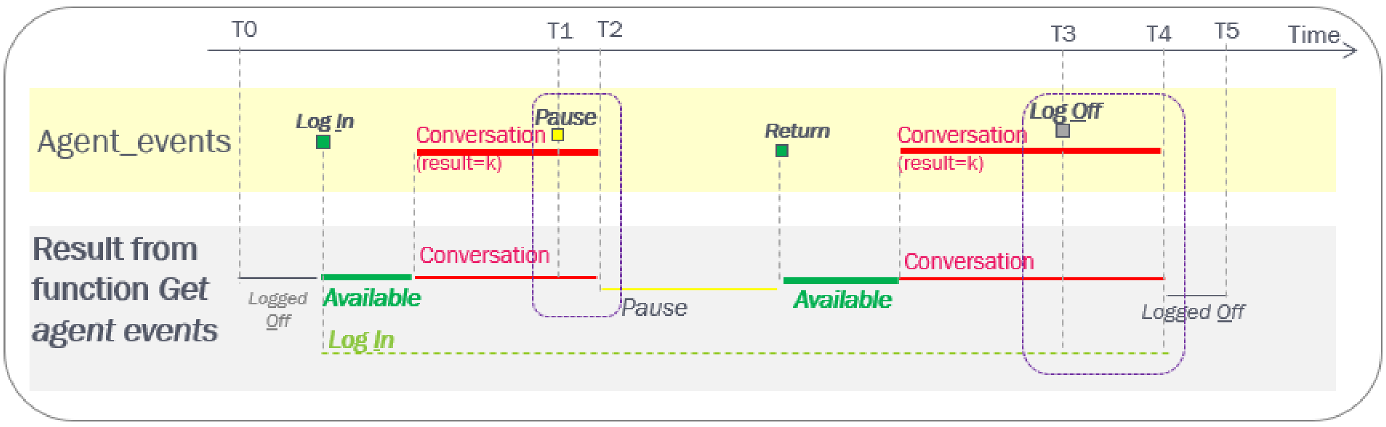
If you add “where event_type not in ('i')” to the query, row number 2 in the result (the login event with adjusted duration) will not be included.

The values in columns adj_dte_start and adj_duration_sec are used to make the next 2 illustrations, but the actual length of each event in the graph is not correct:



What if the agent clicks Pause while connected or Log off while connected?

- If the agent clicks Pause while connected (T1), the agent has status [Paused AND Connected] until call end (T2), and there will be no wrap-up.
- If the agent clicks Log Off while connected (T3), the agent has status [Logged off AND Connected] until call end (T4), and there will be no wrap-up.



New from DB v1.5 in these 2 cases:

- The Pause (Log Off) event's adj_dte_start will be moved from T1 to T2 (from T3 to T4) and its adj_duration_sec will be reduced
- The Log In event's adj_duration_sec will be increased (so that it ends at T4)

So, how do I find the time an agent has been "idle"?

Idle time might be defined as the time an agent is logged on and not in pause and not busy working with a request.

- For agents that only handle phone in Puzzel, one could say that the agent's idle time is equal to the time the agent is in status Ready. This will be the time reported as Available plus the duration for Conversation events not answered, when using the function Get agent events
- For agents that handle Chat, Social and/or Email in Puzzel, this is more complex. Please see chapter about Chat. The short version is that you have to include the conversation events for chat, social and email that the main agent's secondary agents receive, in order to calculate when the agent is actually idle. One agent might have overlapping chats/social/emails, so the idle time is not equal to total time minus sum "speaktime" for all the conversation events for the agent
    - If your agents handle chat, social and/or email in addition to phone in Puzzel, the available events represent the time the agent is logged on and not in pause and not speaking (phone) and not in wrap-up, but possibly connected to a chat/email/social request.

Puzzel agent status change based on Microsoft TEAMS status

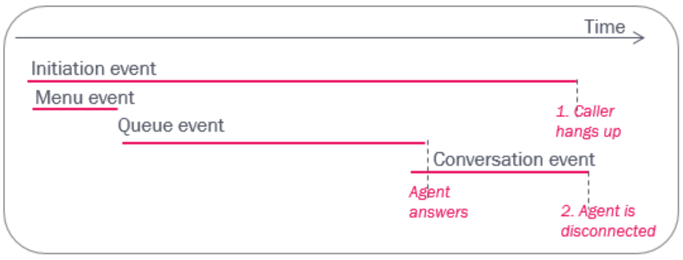
If a Puzzel agent has the property 'Update presence on user from Microsoft Teams through Agent Application' ON, this agent's Puzzel status will be changed from *Ready* to *Busy* if the user becomes engaged in a Teams call.

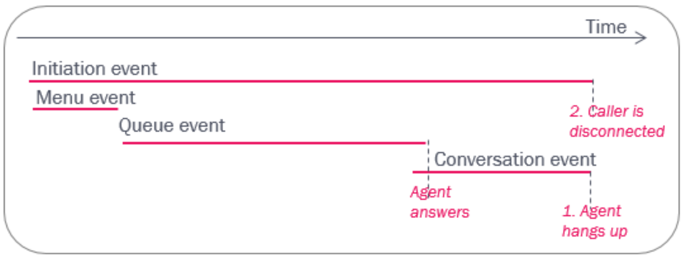
When the user's Teams status is back to Available, the Puzzel status is changed back to*Ready*.

Please note that we don't create any raw data events for such status changes, but the agent's time in Ready status is reset so this affects allocation and which agent gets the next request from queue.

Who hung up first?

When a caller is connected to an agent, the standard behaviour for a Puzzel solution is that when the caller hangs up first, the agent is disconnected, and when the agent hangs up first, the caller is disconnected.





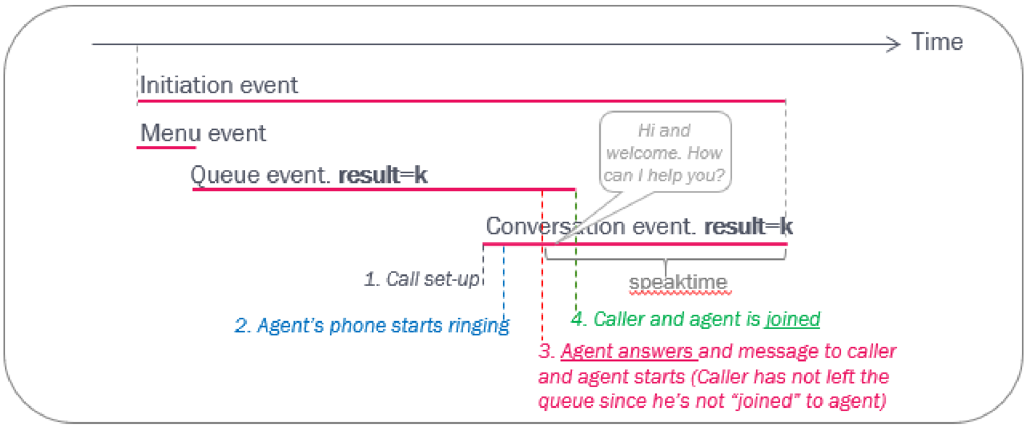
The actual time the caller was disconnected is reflected in the finish time in the initiation event, and the agent's disconnect time is found in the conversation event's finish.

Please note that if the caller and the agent were finished speaking and said goodbye, it's not unusual that the agent hangs up first. And, remember that the connection to the caller's phone or to the agent's phone may be lost due to network problems without the caller or the agent hung up on purpose!

Announcement to caller and agent before join

The default solution is that when an agent answers a call from a queue, the caller and the agent are joined immediately, and both the queue and the conversation event get result=k. The queue event ends when the agent answers (= when the speaktime starts).

If it is configured that a message is played for the caller and the agent before joining caller and agent, the queue event does not end when the agent answers the call, but when the parties are joined after the message is played.

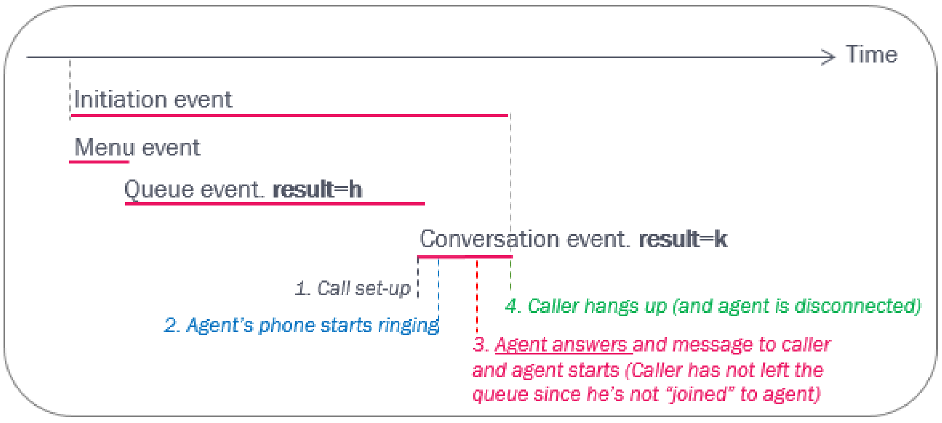

Time
Initiation event
Menu event
Queue event. result=k
Hi and welcome. How can I help you?
Conversation event. result=k
1. Call set-up
speaktime
2. Agent's phone starts ringing
4. Caller and agent is joined
3. Agent answers and message to caller and agent starts (Caller has not left the queue since he's not "joined" to agent)

Hang-up in queue after agent answer (while message is played)

If it is configured that a message is played for the caller and the agent before joining the two parties, the caller or the agent can hang up while the message is playing!

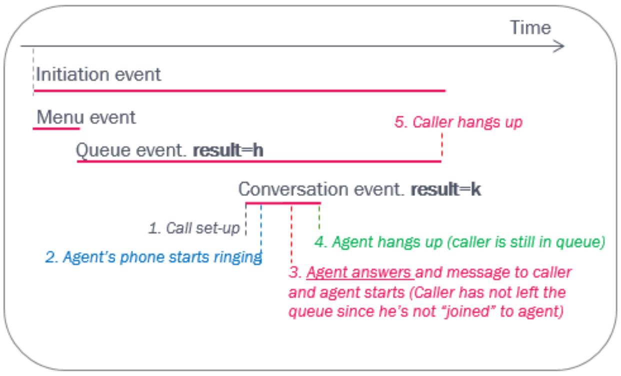
If the caller hangs up while a message is played for agent and caller:

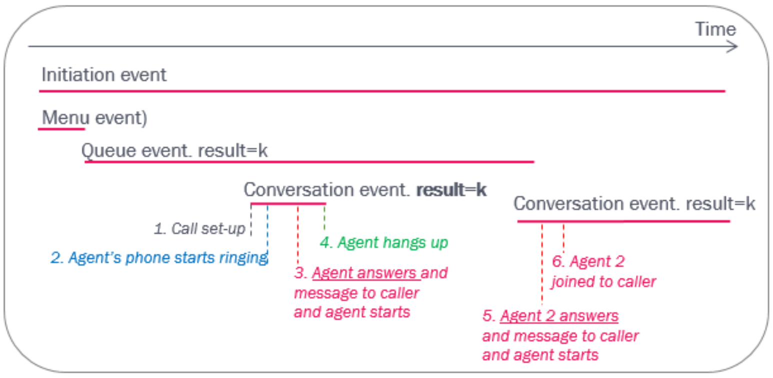
- the queue event is given result=h (since caller was not joined to agent)
- the conversation event for the agent is given result=k.


Time
Initiation event
Menu event
Queue event. result=h
Conversation event. result=k
1. Call set-up
2. Agent's phone starts ringing
4. Caller hangs up (and agent is disconnected)
3. Agent answers and message to caller and agent starts (Caller has not left the queue since he's not "joined" to agent)

If the agent hangs up while a message is played for agent and caller:

- the conversation event for the agent that answered gets result=k, and the caller is returned to queue to wait for a new agent (since he was not joined to agent)
- the queue event's result depends on how the caller later leaves the queue


Time
Initiation event
Menu event
5. Caller hangs up
Queue event. result=h
Conversation event. result=k
1. Call set-up
2. Agent's phone starts ringing
4. Agent hangs up (caller is still in queue)
3. Agent answers and message to caller and agent starts (Caller has not left the queue since he's not "joined" to agent)


Time
Initiation event
Menu event)
Queue event. result=k
Conversation event. result=k
Conversation event. result=k
1. Call set-up
2. Agent's phone starts ringing
4. Agent hangs up
3. Agent answers and message to caller and agent starts
6. Agent 2 joined to caller
5. Agent 2 answers and message to caller and agent starts

Special case

If a caller waiting in queue hangs up just after the allocated agent answered, the agent conversation event will have result=k (answered) and speaktime=0, and the queue event will have result=h (hang-up) since the caller was not joined to the agent.