



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.

# Voice Callback and URS

VCB Configuration

5/8/2025

# VCB Configuration

URS has an option `vcb`, which controls the different aspects of the VCB notification functionality in URS.

## Important

The `vcb` option is dynamic from URS version 8.1.400.39.

The value of this option is in the format of a sequence of numbers:

`n1:n2:n3:n4:n5:n6:n7:n8:n9:n10:n11:n12:n13:n14:n15:n16:n17`, and by default is, `60:90:90:0:20:0:1:25:0:0:150:60:5:0:0:0:300`. The `vcb` option is set at the URS level (though it is possible to adjust it at a call level, see **Step 7** in [VCB Implementation](#)).

The meaning of these numbers is as follows:

- **n1 (AHT threshold**, by default (from 8.1.400.40) 60, before that 30) – when an agent ready event triggers a VCB notification, URS can send the notification immediately or with some delay. If expected time when next agent becomes ready (queue AHT time) is high, then URS can decide to delay the notification to not alert the customer too early. Specifically, if this time (AHT) is more than `n1` seconds then URS delays notification by `AHT/2` seconds. AHT time here is counted for the router's queue (that the VCB call is in). It is the same value as returned by the URS function `RvqData[internal_queue_id, RVQ_DATA_AHT]`.
- **n2 (block call**, by default 90) – logically can be interpreted as the maximum time needed to connect an outbound call with the customer. When the VCB notification for a call is sent, URS might not know what happens with this notification - was it processed somehow, what is the result of this processing (there is no formal feedback event). As a result, the VCB call might remain in the VCB notifiable state and trigger another notification on behalf of another agent, and so on. To prevent a single VCB call from *hogging* all agents once the VCB notification is sent for a call, the call is blocked from sending any other VCB notification for the next `n2` seconds. If, after expiring of this interval the call is not yet routed and is still in the VCB notifiable state, then the next VCB notification can be applied to this call. The other option affecting call blocking time is `n17` (in effect, call blocking time is `n2+n17`).
- **n3 (block agent**, by default 90) – effectively the same as `n2`, but in context of agents. If an agent becomes ready and triggers a VCB notification, this agent can remain ready (no other calls sent to him) and as a result trigger other VCB notifications. To prevent this from happening (not to generate too many notifications) this agent is blocked from triggering any other VCB notifications (for other calls) for `n3` seconds.
- **n4 (multi-URS VCB**, by default 0) – can be 0 or 1 and indicates whether any other URS instance exists that can route a call to the same agents. When the VCB notification for some call is made it is expected that this VCB call is the first in queue. Every URS instance however, knows only its own queue's meaning and other URS instances can have higher priority calls. In such a case, no VCB notifications should be made. In other words, this flag indicates if there are multiple URS instances or not. If the multi-URS mode is set for VCB notifications then it also must be set for regular routing – in other words URS also must have the `agent_reservation` option set to on (if not, this `n4` value will be ignored). If `n4` = 1, then before making a VCB notification (or before routing) URS will try to explicitly reserve the agent triggering the VCB notifications with priority data taken from the VCB call (or the call that is

routed). VCB notifications will be distributed only if/when URS gets a confirmation on this reserving request. The reserving request is made on behalf of the agent DBID and as a result does not interfere with routing reservation. It also means that when a call is selected and routed it might result in 2 reserving requests to T-Server – one VCB related and another regular routing related.

### Important

This flag, if set works even if URS does not participate in VCB. It just causes URS to signal about the call when routing the call. And VCB URSes (if they happen to exist) will use them to compare priorities and adjust the sending of VCB notifications.

**Note:** The above described agent reservation does not work for SIP cluster nodes (it will not accept reservation on behalf of any object except the actual DN). So, when SIP cluster is part of the deployment, ensure that if this flag is set, the reserving request is not sent to the SIP cluster nodes.

#### Since 8.1.400.23

- **n5 (after route queue max**, by default 20) – controls how far URS will go through a queue of calls after a routable call is found. See **Step 2** and **Step 3** in **VCB Implementation**.

#### Since 8.1.400.26

- **n6 (max notifications**, recommended to set to 5 but for compatibility by default is 0) - can be used for tracking and eliminating dead VCB calls. Lost (stuck) VCB calls are those that do not exist in the module accepting VCB notifications and as a result VCB notifications for such calls are just ignored. This can easily result in a small amount of dead VCB calls clogging the distribution of VCB notifications completely. In this scenario, all VCB notifications (or a major part of them) will be for such dead calls. If n6 is not 0 then URS counts for every VCB call how many unanswered VCB notifications in a row was done. The counter is cleared only if URS gets an external message (it does not matter what the message is about) for this call – such as a web request or a message from ORS. If the counter however, manages to reach n6 then such a call is temporarily removed from the pull of VCB calls for 20 minutes. If, after expiration of this time, the counter is still not cleared, then one more VCB notification for this call is allowed and if it still results in no action, the processing of this VCB call is terminated inside URS.
- **n7 (use any agent**, by default 1) – allows using any ready agent for for triggering a VCB notification. The routing strategy (including those processing VCB calls) can impose many different extra conditions on agents the call can be routed to (different thresholds and so on). If n7=0, then URS checks all those conditions for VCB calls to become eligible for VCB notifications (in a general scenario, it is not possible to predict when the customer will answer the call and some other agent will become ready – whether or not all those conditions are satisfied).
- **n8 (check queue**, by default 25 from 8.1.400.40, before that 50) - controls the definition of the *surplus of high priority VCB calls* case. See **Step 3 (Use Case C)** in **VCB Implementation**.
- **n9 (jump priority**, by default 0) – allows to automatically boost call priority by n9 when VCB calls become routable (customer answers the call).

### Important

A consequence of jumping priority to be aware of in versions before 8.1.400.40 is that, answered VCB calls might be concentrated at the beginning of the queues (as having moved up in priority). This means that a search of VCB notifiable calls will effectively be moved to post routing which is limited by the n5 parameter of vcb option (by default 20). Taking into account the existence of not answered calls, that might leave very little room for finding the next VCB calls to dial. In other words, jumping priority slows down the notifications rate.

- n10 (**one blocker**, by default 0) – deprecated, set it to 0.

### Since 8.1.400.37

- n10 (**secondary notification**, by default 0) – If set to 1, then the agent ready event can trigger one extra VCB notification (if no call was routed to the agent). See **Step 6** in [VCB Implementation](#).

### Since 8.1.400.40

- n11 (**dialing rate**, by default 150) – Defines *acceleration* factor of dial out rate. Value specified as percentage of *minimal* rate (one agent ready event triggers one notification). Valid range is 100-300. See **Step 7** in [VCB Implementation](#).

**Note:** This option might conflict with the DialOutSuccessRate function. It is recommended not to mix them - in a scenario when dial out rate is controlled with the function for some calls and with the option for others, might result in a quite unstable dial out rate.

- n12 (**over dial by waiting time**, by default 60) – Defines condition of over dial situation. If going through a queue of calls waiting for an agent, a former VCB call (that is, call already connected with the customer) will be detected as waiting for more than the specified time and URS will assume it has an over dial situation. If the value is set to 0, then detecting of over dial situations by time is disabled. See **Step 7** in [VCB Implementation](#).
- n13 (**over dial by calls**, by default 5) – Defines condition of over dial situation. If going through a queue of calls waiting an agent, the specified amount of former VCB calls (that is, calls already connected with the customer) will be detected and URS assumes it has an over dial situation. If the value is set to 0, then detecting of over dial situations by number of calls is disabled. See **Step 7** in [VCB Implementation](#).
- n14 (**over dial rate**, by default 0) – Defines dialing rate to be used if an over dial condition is detected. In an over dial situation, URS can totally suspend dialing (if option value is 0) or still consider making one VCB notification. The option defines the probability (in percentage) that the one notification will still be made. This option is intended to make the reaction to an over dial situation as efficient as possible, and reduce the rate of VCB notifications instead of suspending them completely. Valid values for this option are 0 to 100. See **Step 7** in [VCB Implementation](#).
- n15 (**reserving agent AHT**, by default 0) – slow queues case. If positive, it defines the average handing time threshold, on exceeding which URS in addition to sending a dial out notification will also block/reserve the agent after the VCB call. See **Step 7** in [VCB Implementation](#).
- n16 (**AHT gap history**, by default 0) – slow queues case. If positive, it defines the timeframe (in

seconds) within which URS will check for gaps in holding time for the internal queue the VCB call is in. If a gap is detected, then URS, in addition to sending a dial out notification will also block/reserve the agent after the VCB call. See **Step 7** in [VCB Implementation](#).

- **n17 (rest time**, by default 300) - used to extend call blocking time (n2) after sending a VCB notification. Normally it is expected that if dialing to customer fails (n2 expired) then the VCB application will place the VCB call into a route delay state for some extra time. In addition to this processing by the VCB application, an extra rest time for the call can be specified here. In effect, if sending a VCB notification results in no action, then the ext notification for this call will be sent not sooner than  $n2 + n17$  seconds.

**Note:** Since 8.1.400.40, there is a possibility to specify the above listed VCB configuration parameters with separate options. The VCB options are specified in a dedicated vcb section. URS first checks the default/vcb option and if found sets the VCB parameters accordingly. In addition, URS will check the URS application for the vcb section and if found will try to set the VCB parameters with the specified values. The correct names of the options inside the section are as follows:

vcb_aht_threshold = n1
vcb_block_call = n2
vcb_block_agent = n3
vcb_multi_urs = n4
vcb_after_route_max = n5
vcb_max_notifications = n6
vcb_any_agent = n7
vcb_check_queue_max = n8
vcb_jump_priority = n9
vcb_secondary = n10
vcb_default_rate = n11
vcb_overdial_wait_time = n12
vcb_overdial_calls = n13
vcb_overdial_rate = n14
vcb_reserving_aht = n15
vcb_aht_gap_history = n16
vcb_rest_time = n17