



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.

Universal Routing Reference

Estimated Waiting Time Improvement and URS Web API

4/16/2025

Contents

- 1 Estimated Waiting Time Improvement and URS Web API
 - 1.1 Method `urs/call/connid/lvq`
 - 1.2 Method `urs/call/connid/query`
 - 1.3 Function `RequestRouter`
 - 1.4 `InVQWaitTime` Enhancement

Estimated Waiting Time Improvement and URS Web API

Universal Routing improves its capability of calculating the estimated waiting time for a routing target.

- Starting with URS 8.1.400.25, you can use URS Web API methods to calculate a more precise estimated waiting time. With this release, the following two Web API methods support the improved estimated waiting time functionality: **lvq** and **query**.
- The existing **InVQWaitTime** function is enhanced to no longer require the association of agents with virtual queues and can be used in an environment with multiple URSs, queues, and multi-skilled agents.
- Starting with IRD 8.1.400.21, a new function, **RequestRouter** is available in IRD's Function object, which simplifies calling the Web API methods from routing strategies.

Method urs/call/connid/lvq

The lvq method allows more control over how estimated wait time is counted. The lvq method can also be applied to groups of virtual queues. If a virtual queue name ends with .GQ URS counts estimated waiting time and other parameters for the entire collection of virtual queues.

When the lvq method is invoked with the aqt parameter, the calculation includes:

- stat—identical with value counted by the InVQWaitTime function.
- urs2—Similar to the InVQWaitTime function but URS itself counts AverageHandlingTime. It does not matter whether or not a virtual queue is configured with an Agent Group.
- urs—Default value, Like urs2, but instead of AverageHandlingTime, URS uses average quitting time which is counted based on last 32 calls distributed from the queue.

Returned values for the lvq method depend on the input parameters supplied. Samples:

```
urs/call/connid/lvq/VirtualQueueName?aqt=stat  
urs/call/connid/lvq/VirtualQueueName?aqt=urs  
urs/call/connid/lvq/VirtualQueueName?aqt=urs2
```

Important

If aqt=urs2, URS calculates an agent's average handling time based on the agent's CurrentState/CurrentTargetState statistics. AHT is calculated based on the last 10 calls (for every media) that an agent processes. If aqt=stat, URS calculates an

agent's average handling time using a sliding window of 10 minutes.

To view input parameters and output information: With URS running, open a browser and run the help method for lvq. The help method is described in the [Universal Routing 8.1 Reference Manual](#), Appendix C, "Supported Methods." Example: `http://host:port/urs/help/call/lvq`

The lvq method can be called from a routing strategy with function [RequestRouter](#).

Method urs/call/connid/query

The query method is described in the [Universal Routing 8.1 Reference Manual](#), Appendix C, "Supported Methods." This method can be helpful when the use of virtual queues is not appropriate or not possible for calculating estimated wait time. For example, a virtual queue could include calls that might wait for totally different targets or have different thresholds. This method uses router's queues instead of virtual queues (all calls in router's queues wait for the same targets and have the same thresholds). Internally this method uses URS function RvqData to get aggregation information about all router's queues where a call resides.

When the query method is invoked, the calculation includes:

- `min_rvq_ewt`—Minimal estimated waiting time among all router's queues call is counted with using URS-provided `AverageHandlingTime`.
- `min_ewt`—Minimal estimated waiting time among all calls in router's queues is calculated by URS based on average quitting time.

Notes:

- Calculation of `AverageHandlingTime` that URS provides for router's queues does not consider multi-skilled agents or support multiple URSes.
- The query method has no input parameters.

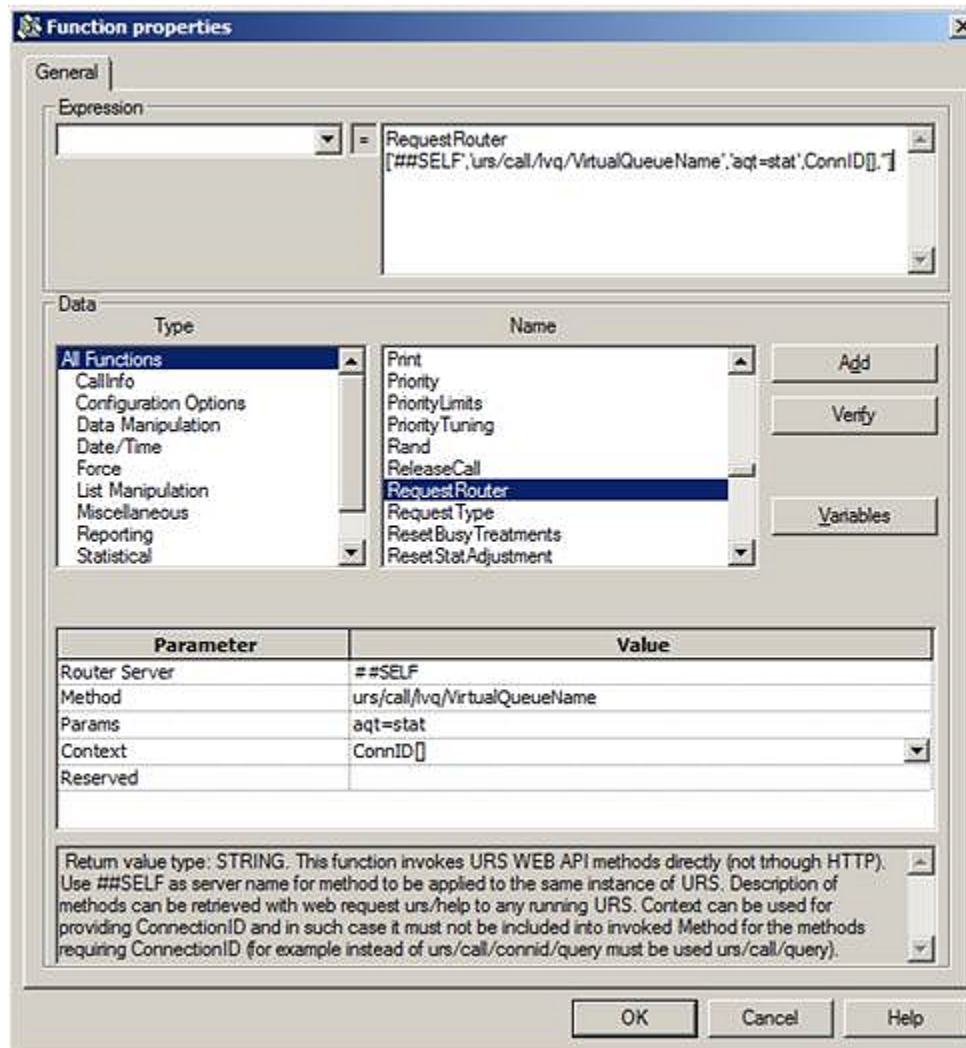
This method can be called from a routing strategy with function [RequestRouter](#).

Function RequestRouter

This function simplifies using URS Web API methods from routing strategies. It allows you to directly invoke URS Web API methods, which can be any supported methods. Example:

```
RequestRouter['##SELF', 'urs/call/lvq/VirtualQueueName', 'aqt=stat', ConnID[], '' ]  
where STRING RequestRouter[STRING RouterServer, STRING Method, STRING Params, STRING  
Context, STRING Reserved]
```

A sample IRD Function properties dialog box using the lvq method is shown below.



Parameters:

Router Server: Uses ##SELF as the server name for the method to be applied to the same instance of a running URS. Default value: '##SELF'.

Method: With IRD release 8.1.400.21, supported methods for use with this function are urs/call/lvq or urs/call/query.

Params: If the method must have, but does not contain parameters, they can be provided separately here.

Context: If the method must have, but does not include an Interaction ID, it can be provided here. Can also be used for providing a ConnectionID, in which case it must not be included into invoked methods requiring ConnectionID. For example, instead of urs/call/connid/query, URS just uses urs/call/query.

Reserved: Not used in IRD release 8.1.400.21.

Return Value Type: STRING

InVQWaitTime Enhancement

Starting with URS release 8.1.400.25, the InVQWaitTime function is enhanced to no longer require the association of virtual queues with Agent Groups.

- If a virtual queue is associated with some Agent Group, then URS uses Stat Server's $\min(\max(\text{StatLoadBalance}, 0), 10000)$ statistic for the virtual queue and adjusts it relative to call position in the queue.
- If a virtual queue is not associated with an Agent Group, then URS calculates an AverageHandlingTime for all agents from the virtual queue that URS considers as a target and multiplies the calculated result by the call position in the queue. While calculating an AverageHandlingTime, URS attempts to consider multi-skilled agents and multiple URSes.

Important

For more information on Expected Wait Time (EWT) and its related statistics, functions, and methods, refer to the [White Paper on EWT](#).