Context Service Connection Timeouts

The Context Service client provides **REQUEST_TIMEOUT** connection property that significantly affect the user experience for your customers and agents. **REQUEST_TIMEOUT** defines how long the client waits for a response from Context Service. Defaults 5000 ms.

Ideally, you set REQUEST_TIMEOUT to a value that is long enough that the client does not frequently hit this limit before receiving a reply from Context Service, but not so long that callers hear long periods of silence from the IVR. Context Service is a cloud service that runs over the Internet, and the latency of the Internet can be highly variable. However, latency generally has a floor value based on your physical proximity to the data center on which your Context Service instance is hosted. You can roughly determine your floor value by pinging your Context Service instance at the times you experience the lowest call volume. Your REQUEST_TIMEOUT must never be lower than this floor value. Setting REQUEST_TIMEOUT lower than this value typically triggers timeouts and initiates a retry.

If Context Service does not respond on that first retry, then the request fails. If there is an issue with your connection, then you are potentially doubling the wait defined in REQUEST_TIMEOUT. The default REQUEST_TIMEOUT is 5000 ms (5 seconds). This combination results in a 10 second wait (`wait time = REQUSET_TIMEOUT * 2`). In a worst-case scenario, wait time can be significant if there are network issues between your client and Context Service and your timeouts are set too high.

In general, typical requests within the same geographical area take from 100ms to 300ms. However, your network environment, switching latency, and location in relation to Context Service instance can increase the response time from Context Service.

Your service quality target ultimately defines how high REQUEST_TIMEOUT is set above the floor latency value. Setting the value too high results in extended waits for the caller or agent when Internet latency is high. Setting the value too low initiates retry requests that increase the wait in an attempt that can ultimately fail during times of high latency.

You can improve the customer experience of waiting while the client is accessing Context Service by notifying the customer that you are looking up information. For example, you can play prompts such as "Wait a moment while I access your account details". If a timeout occurs and a retry is attempted you can play a prompt, such as "I'm still accessing your account details." You could also opt to play MoH during the wait times to prevent silence on the line.

Inevitably connections to Context Service can fail, possibly due to high Internet latency or connections issues to the Internet itself. In those cases, your IVR scripting must account for a failed connection attempt to the Context Service. Your scripts must be able to route to an agent (or continue with self-service) without the benefit of Context Service data.