

# FAQ

## How many profiles should I request in each call to `jsSendMultipleProfileRequest()` / `BeginGetQueuedProfiles()`?

In sync modes, the scan heads buffer acquired profiles until they're read out by the controlling computer. In high-speed scanning applications, the readout is typically done in a loop that calls the above functions repeatedly and processes the resulting profiles. Depending on the scanning application's load, that loop can potentially be slower than the acquisition speed, leading to profiles being accumulated in the scan head. The goal is to keep the amount of buffered profiles low, so there's not a long time difference between when a profile was measured and the time it's processed. As a rule of thumb, a number between 1 and 20 is a good starting point. Keep in mind that if the readout loop is faster than the acquisition speed, requesting more profiles from the head than are buffered is perfectly fine – the head just will return an empty list.

The opposite situation is more difficult – if you read out profiles slower than they are measured, the buffer can fill up and eventually overflow, leading to dropped profiles and sequence errors. Reading larger numbers of profiles is also somewhat inefficient from a networking perspective.

As a simple check, you can compare the fields `SendLocation` and `Location` in the resulting profiles. The difference between these fields should remain constant over time. If `(SendLocation-Location)` grows steadily over time, it is a sign that your readout loop is not keeping up with the measurement process and eventually you will overflow the buffer. In this case you must speed up the readout loop (to get fewer profiles more frequently) or request more profiles at a time (to get more profiles less frequently).

In practice it has been shown that a well-designed threading model on the readout loop is reliably able to keep up even at high scan speeds if the processing of profiles is offloaded from the readout loop itself. The Developers Kit documentation has examples of how to de-couple acquisition and processing.

## My scanner shows up in JSConfig, but not in JSdiag. What's wrong?

Most likely, you have an IP address conflict: two devices on this subnet are assigned the same IP address. JSConfig will warn you if two scan heads have the same IP address, but it can not detect if another device or the control computer have an IP address assigned to a scan head. This can lead to unstable behavior, with heads disappearing randomly or connections dropping. To isolate the problem, you can remove devices from the network, until the problem disappears, then checking the IP settings on the device that were removed.

A common situation is that the control computer's IP address overlaps with the block assigned to the scan heads. In this case, the conflict scan head will show up in JSConfig, (seemingly) respond to pings but fail to connect in JSdiag. You must change the IP address of the control computer not to conflict with a scan head.

## No scan heads show up at all in JSConfig.

Most likely, there is a network firewall active on your computer, silently discarding all network packets that the scan heads send in response to a discovery request sent by JSConfig. Please see the note in the [JSConfig](#) manual on how to disable the firewall temporarily. JSdiag and custom software communicating with the scanners will work fine, as they do rely on a different network mechanism.