



Tip of the Week :

- Setting/Using Link Queues and CA_SYNCNOTIFY

[Client-Side Queues]

- Normal case:
 - 1. Data arrives from server
 - 2. Data is displayed by the application (or otherwise processed/written to disk)
 - 3. Wait for the next data set

[Client-Side Queues]

- 'burst' case:
 - 1. Data arrives from server
 - 2. Data is displayed by the application (or otherwise processed/written to disk)
 - But the next data set has arrived while I was in step 2!
 - What to do with it?

Client-Side Queues

- At the socket level:
 - RECV Buffer queues hold on to unread data if there is room
 - No room ?
 - UDP: data is discarded
 - TCP: both end points wait until it's read
 - What can the applications Programmer do?
 - Can set the RECV Buffer space (default 64K)

[Client-Side Queues]

- If data-receive event notification is asynchronous (*in C, VB, LV it is by default*):
 - Unhandled data is queued up to the queue depth before being discarded.
 - Default depth = 10
 - Can set this via API:
 - `void SetClnRecvQueueDepth(int depth);`
 - global scope -> all links
 - `void SetLinkQueueDepth(int linkId,int depth);`
 - link specific
 - Good for handling 'bursts' or brief irregularities

Client-Side Queues

- 'worst' case (pathological):
 - 1. Data arrives from server
 - 2. Data is displayed by the application (or otherwise processed/written to disk)
 - 3. But the next data set has arrived while in step 2 and is queued.
 - 4. 1-3 always true => queue is full
 - Most recent data is either rejected or the last to be displayed by the application!
 - e.g. Application gets 10MB of trace data @ 5 Hz but takes 1000 ms to display it -> always displaying something that is 5 seconds old.

[Client-Side Queues]

- Solutions:

- Fix the display problem !
- Set the queue depth to 0
 - At least display the most recent data you can!

- Questions:

- How important is it not to 'miss' anything?
- Is the 'best effort' to display the most recent data okay?

Client-Side Queues

- Queue-depth = 0 Side effects:
 - If asynchronous notification, data could be overwritten with latest values if still handling the previous notification!
 - May or may not be a problem
 - Can add the CA_SYNCNOTIFY access flag.
 - Event notification occurs on data arrival (not deferred).
 - But: synchronous links inside of callbacks not possible!
 - Synchronous links (ExecLink()) use CA_SYNCNOTIFY by default.