

This feature can be useful for instant raytracing of fast-moving objects. Normally, a physics proxy will be one frame ahead of the render geometry due to multithreading, which means aiming at the render proxy can miss. Also, *pe\_status\_pos* will be guaranteed to return the same coordinates that were set by a preceding *pe\_params\_pos*, even if the command was queued and hasn't been fully executed yet.

This feature optionally keeps a second set of top-level entity coordinates (pos+quat) that is synchronized to the main set when a logged poststep event is sent (in the main thread), meaning it's in sync with the main CEntity coordinates. World queries (RWI, PWI, GEA) can optionally do tests using these coordinates. Also, *pe\_params\_pos* requests modify them instantly, and *pe\_status\_pos* can optionally read them instead of the main coordinates (*meaning that SetParams(pe\_params\_pos); GetStatus(pe\_status\_pos); will return the updated position even if SetParams was queued*).

To enable double buffering for a specific entity, *pe\_params\_pos.doubleBuffCords* must be set.

For character skeletons, it must be set for both *CLiving host* and the character itself.

To use these coordinates in RWI, PWI, and GEA, *ent\_use\_sync\_coords* must be set in *objtypes*. To use them in *pe\_status\_pos*, *status\_use\_sync\_coords* must be set in flags. By default sync coords are not enabled and not used.