When creating LocalCalCoords, we do the skipped day check as if the frame of reference is wall, even though this has the potential of giving a false positive. This is because the skipped days begins and ends at midnight wall time, so a valid standard or universal time can occur on a day that would have been skipped in wall time. The alternative is to only check skipped days when the frame is wall. In any case, this should not be a problem as LocalCalCoords only should be used by internal functions, for example to create the boundary times for the PeriodTimeZone array. The problem can be avoided if the LocalCalCoords is created via an offsetLocalCalCoords call, which adjust the calendar, appropriately.

When creating UTCDatetime, the check is performed assuming wall time. This is technically correct, since for UTCDatetime, all frames of reference are the same.

When creating LocalCalCoordsDT the skipped day check can be performed properly. The skipped period should not be defined in terms of days, but rather in terms of LocalCalCoords, similar to the boundaries for PeriodTimeZones. We can initialize the LocalCalCoordsDT without checking for skipped days (which means the current createDateCoords call cannot be used, since it calls checkDateElements). We then can determine the last Julian instant and first Gregorian instant as LocalCalCoords in wall time. If the LocalCalCoordsDT is in wall time, this is adequate. If not, we need to create the time zone fields for these boundaries, so they can be translated to the appropriate frame of reference. The initialized LocalCalCoords now can be compared to those values to check for skipped days. The same approach can be applied to LocalCalCoords. This will work since we can assume the Julian to Gregorian transition does not occur at an ambiguous time, so the added information associated with the LocalCalCoordsDT is not necessary.

\*\*\*\* Write a function that takes a calendar, a time zone and a frame of reference, and generates the boundaries as LoalCalCoords \*\*\*\* Make sure offsetLocalCalCoords doesn’t defeat this by checking for skipped days. Perhaps we make offsetLocalCalCoords ignore the calendar adjustments, while offsetLocalCalCoordsDT respects them.