Clarification Request

References: BTL Test Plan 12.0 test 9.31.1.1 (and maybe others)

Date of BTL-WG Response: October 24, 2013

Background:

When issuing a (UTC)TimeSynchronization-Request to an IUT, some implementations need a considerable time for calculating internal changes. Due to things like recalculating schedules and other time dependant states the required time may even be significantly higher than normal InternalProcessingFailTime. There seems to be no Fail Time defined to take this time delay into account. Test 9.31.1.1 issues a TimeSync and does a ReadProperty on LocalTime and LocalDate immediately after that.

The testscript for 9.31.1.1 in BTF currently (as of BTF 12.0.1.42) does contain a WAIT InternalProcessingFailTime by mistake. Some testers specified a higher FailTime just to pass that test. When doing the test manually the problem will likely stay undiscovered because the tester is typically slow enough in typing to avoid the problem.

In a number of tests for scheduling the problem is hidden because the additional ScheduleEvaluationTime covers delays in the IUT.

Question:

Should a vendor defined TimeSyncFailtime be introduced that should be applied in all tests, that issue TimeSynchronization-Request (or UTCTimeSynchronization-Request) before evaluating the result of the request in the IUT?

Response:

Coding in an automation tool of a WAIT Internal_Processing_Fail_Time after the IUT receives an unconfirmed request is a good idea. As in all software, execution takes some time and a device that performs the next step not immediately, but within a reasonable time, should not be considered Failed. TimeSynchronization_Fail_Time is a reasonable concept and BTL will put it in our future items to consider as we review the test plan.