Evaluation of Latent Fingerprint Technologies
Phase 2 - FAQ

#	Question	Answer
1.	Are you going to disclose the distribution of Rotational Tolerance in the Phase II dataset.	No information will be provided before the start of Phase II regarding the distribution of fingerprint orientation (e.g. rotational variance) in the Phase 2 dataset.
2.	Do you intend to assign +/- 180 degree variance for all latent searches on Phase 2?	Perhaps. This was the case in Phase 1. For Phase 2, we may also choose to enroll latents with the parameters 'orientation' and 'range' set to 90 (upright) and 180 (+-180) respectively, and therefore matchers who are sensitive to orientation should be prepared to handle this general case where orientation is unknown (i.e. the full range 0 to 359 is possible). Note that even if the orientation is specified more precisely than +-180 from upright (e.g. +-30 from upright), all rotational variance information conveyed to the SDK is informative only, and not instructive. SDKs may optionally choose to utilize this information in any way they see fit. For example, a matcher could ignore any input regarding orientation, and choose to determine orientation autonomously.
3.	If a fingerprint examiner knows the correct orientation of a latent, he/she would try to re-orient it in a near up-right position, and specify an uncertainty value around it, no?	One goal of ELFT is "lights out" operation, where a human examiner has little or no involvement. Thus, in the general case the orientation of the latent will be unknown (i.e. +- 180 degrees variance from upright). That said however we want the capability to specify a range of rotational variance in cases where it is known, for testing purposes. Although, this information may not be available at the time of Phase 2 testing.
4.	Please clarify that if by using threading, the time requirement for matching will be lower than the specified 0.25sec/record (or do we now have twice as much CPU time if we use 2 threads)?	The timing requirements are the same however use of threads will be noted in the report. Bounded by Amdahl's Law, 2x speedup is possible (sometimes even greater due to cache behavior, etc.). Thus, any remaining cycles may be exploited if worthwhile. The goal here two-fold: (i) better utilization of our dual-core cluster; and (ii) quicker testing when realized speedups outpace increases in algorithmic complexity.
3.	Will the test machines will be 'clean' making	Each SDK will have the complete dual-core

	most all of the 2GB of memory available to the match process. If operating system and your test software memory could be given as a baseline it would help to let us know how much memory will be available for template storage.	system at its disposal. (Same as in Phase I). At least 2GB RAM will be available, allowing for worst-case overhead.
4.	In the 5th paragraph of "description" on page 5, it says: "Non-fatal error conditions shall be tolerated and shall not result in premature halting "i.e., non-completion of background enrollment) and a non-zero return code." (1) Is this implying that the function shall return a 0 (namely, success)? (2) Is it that any errors (strictly speaking, error conditions) other than ones listed in the list on page 10 are considered as "non-fatal"? (And thus should return 0, assuming (1) is correct?)	(1) In the November 7 revision of the API we have modified this paragraph to the effect that documented (by the SDK provider) non-zero return codes are optional in the non-fatal error cases specified, for the purposes of providing an informational warning. So in other words, the function must tolerate the non-fatal error condition and complete its operation, and upon return it may optionally warn the caller of the occurrence. But it is not required to do so and may return 0 (Successful). In the nonzero case a detailed warning message could be written to the error_msg parameter. (2) No, the conditions listed there are a minimum required set of non-fatal error conditions. Robust SDKs may optionally add other non-fatal conditions to this set. The error codes listed in 2.4 may reasonably be interpreted as being "fatal" errors. However, robust SDKs may optionally tolerate some of these to the maximum extent possible, though they are not required to do so. Note that the error conditions listed in 2.4 are a minimum set, and are not an exhaustive list. It is anticipated that other conditions could be detected and signaled by the SDK with a non-zero return code and if so they must be documented.
5.	The "0...N-1 inclusive" (para 3, p.9 of the Phase II API doc) range seems to be incorrect. Per FAQ 41, this range should have been 1...N. Does this still hold? Please confirm.	Yes, and that error has been corrected in the November 7 revision of the Phase 2 API.
6.	Per your response to FAQ 42, the upright position corresponds to 0 degrees. But, in the 4th para (p.7 of the API), the upright corresponds to 90 degrees. Please clarify.	FAQ 42 is deprecated and 90 degrees specifies upright for the Phase 2 API.