Dynamic binary translators are programs that translate binary programs from one machine to another. The translation is done on the fly, so performance is a major issue in this kind of system. Identifying and optimizing hot traces is a way to achieve more performance, and also to compensate for the translation overhead. Aggressive optimizations need precise data/control-flow information about the code, otherwise they will be conservative and less effective. In this paper, we measure the amount of additional data-flow information one can obtain by going beyond hot trace boundaries into non-frequently executed (cold) code. We show that in some cases, as in liveness analysis, one can considerably improve the information available, thus creating more opportunities for trace optimization. Moreover, the amount of additional data-flow information decreases very fast as one departs from trace boundaries, limiting the overhead imposed by thecoldcodeanalysis.