Implementing Checks and Signals in HRS: Improving Readability and Usability

By Jason Ostergren, Morshed Bhuiyan, Bernadette Jones and Helena Stolyarova.

Background: The current main HRS instrument contains more than 700 signals, a significant number of which use different wording dependent on interview mode: CAPI or CAWI. In addition, the introduction of the Spanish CAWI interview option for HRS2028 required implementing all signals in Spanish, effectively doubling the amount of language text needed for a given signal. In previous waves, each signal was implemented separately as a SIGNAL or CHECK statement after the relevant field in the RULES section of the source code. This approach required separate maintenance and translation for individual signals, many of which were similar in nature.

These challenges motivated the development of a new approach to implementing signals in HRS, with a focus on improving readability, reusability, and translation maintainability. This paper describes the philosophy behind this approach, as well as the challenges encountered during its implementation.

Readability: Moving signals out of the RULES section improves code readability and follows the main HRS programming philosophy of keeping text manipulation separate from the flow logic. This separation makes the code easier to understand, maintain and review.

Reusability: The majority of signals check for similar conditions, such as date validity or the reasonableness of dollar amounts. This similarity allowed us to group signals into categories and create procedures for each category. By parameterizing the signal procedures, information specific to each individual signal can be passed as a set of parameters, significantly reducing programming and translation efforts.

Translation maintainability: Signal texts are implemented as constants and stored in separate files, allowing translations to be managed independently from the source code. Since signal procedures are parameterized, translators only need to translate each signal text once, while variable information such as dates, amounts, or other field-specific values can be inserted dynamically as fills. This approach reduces duplicate translation efforts, improves consistency across interview modes and languages, and simplifies future maintenance and updates.