

DEPARTMENT OF LABOR
Bureau of International Labor Affairs
Solicitation for Cooperative Agreement Applications
[SCA-14-23]

Amendment One

Child Labor Beneficiary Monitoring Toolkit

AGENCY: Bureau of International Labor Affairs, Department of Labor

ACTION: Amendment to SCA-14-23

SUMMARY: The Bureau of International Labor Affairs announced on September 18, 2014 the availability of funds and Solicitation for Cooperative Agreement Applications (SCA) for one or more cooperative agreements to develop a Child Labor Beneficiary Monitoring Toolkit, to be awarded through a competitive process.

This amendment to the SCA adds a new appendix (Appendix H, **attached below**) entitled “Baseline Requirements for a Direct Beneficiary Monitoring System Prototype” and revises references to this appendix in Section I of the SCA.

The document is hereby amended as follows:

Section I. Funding Opportunity Description (new language in bold)

Old Text:

- Prototype Electronic DBMS system (page 5)

Applicants must describe how they will refine the list of requirements presented in Annex D, DBMS prototype baseline requirements, and how they will develop, test (in collaboration with two to three volunteer grantees), and refine a prototype DBMS system based on these requirements.

New Text:

- Prototype Electronic DBMS system (page 5)

Applicants must describe how they will refine the list of requirements presented in **Appendix H, Baseline Requirements for a Direct Beneficiary Monitoring System Prototype**, and how they will develop, test (in collaboration with two to three volunteer grantees), and refine a prototype DBMS system based on these requirements.

Section I. Funding Opportunity Description (new language in bold)

Old Text:

- Desk Review (page 6)

Applicants must also review Annex D (Baseline requirements for a DBMS prototype).

New Text:

- Desk Review (page 6)

Applicants must also review **Appendix H, Baseline Requirements for a Direct Beneficiary Monitoring System Prototype.**

Section I. Funding Opportunity Description (new language in bold)

Old Text:

- Software/Database Developer (page 9)

Excellent command of one of the technologies listed in Annex D (Django/Python, php, or Ruby on Rails, as well as a database management system (MySQL)

New Text:

- Desk Review (page 9)

Excellent command of one of the technologies listed in **Appendix H** (Django/Python, php, or Ruby on Rails, as well as a database management system (MySQL)

Appendix H

Baseline Requirements for a Direct Beneficiary Monitoring System Prototype

Following are initial identified baseline requirements for the proposed DBMS prototype. Requirements are subject to change, and additional requirements are expected to be added as work on the project progresses.

1. The system must be designed to be accessible via a web-based interface, and the records held in a database. The system will be hosted by the organizations implementing OCFT-funded child labor elimination projects.
2. The system must have two types of users: grantee and grantor (OCFT). The system must have authentication (login) and authorization (access to different records must be authorized under defined access control rules depending on the user identify and user role).
3. When accessing the system, grantee must see a menu where they can do at least the following three things:
 - a. Create/edit/view a child record. This child record is held in a specific table (the CHILD table), with each record containing a unique ID field. Whatever demographic information required must be held in separate fields (DOB, address, etc.).
 - b. Create/edit/view a household record. This household record is held in a specific table (the HOUSEHOLD table), with each record containing a unique ID field. The table includes a column listing all children in the household by ID number. Other required demographic information is held in separate fields (address, number of household members, etc.)
 - c. Create/edit/view a status update record. This status update record is held in its own table (the STATUS table). Each status record has its own unique ID field (referencing the unique child or household ID to which it refers). It also contains whatever other fields that are needed for describing the status update (period start date, end date, status code on work and education status, notes field, etc.).
4. The system must include a menu from which users may make pre-determined queries such as to find and review child or household progress, per time period, per project/grantee, per country, etc.
5. The system must provide the option of random assignment of children and/or households entered into the database (to facilitate comparisons across different groups, should the need arise).
6. This system will be used as a prototype for grantees to build their own system, which they will self-host. As part of this prototype, DOL will also require the grantee that develops this prototype to develop specifications for the format and content of the data that will be provided to DOL for aggregation across grantees. These specifications will be developed as part of the development of the prototype. A tool to be used by DOL to aggregate data across all grantees will also be built as part of the prototype. The aggregation tool to be used by DOL should support making queries and building specific tables for internal DOL reporting purposes, and should have a user-friendly graphic user interface. This querying tool and interface should be prototyped as well as part of this effort.

7. The entire system must be built using free/open source software components, such as the following:

- a. The operating system should be some variant of Linux (e.g. Ubuntu).
- b. The database management system should be an open source relational database management system such as MySQL.
- c. The front tier (the web-based interface) and the middle tier (or middleware, i.e. the logic/'glue' that implements the various functions, connects to the database, etc.) should be one of many possible frameworks/languages, in order of preference: php, Python, Django, or Ruby on Rails.

8. The system must be designed to be held and run on an internet service provider, as they already offer the infrastructure needed to run databases such as MySQL and scripting languages like php and Python/Django.

9. The software must be developed with strong cyber-security requirements in mind. In addition, the software must provide tools for users to protect personally identifiable information (PPI) (e.g., means of encryption, access restrictions by user type).

10. If problems with the software are identified by OCFT after its delivery and initial use, the grantee will be required to address these problems.