

ELEKTRA Generating Questionnaires (iFORM)

This document provides technical information to developers of software supporting data provision through KSH-ELEKTRA, on how to create correct import files.

Using KSH-ELEKTRA is based on filling and submitting individual questionnaires. Each questionnaire instance contains the set of information on an identifiable reporting unit of a data provider, for a particular survey and time period.

The layout of a questionnaire in the Data Provider Client, the description of the data to be collected (type, set of values etc.), the validation rules, as well as the references to the help pages of the questionnaire are included in a single file called the **questionnaire template**. Questionnaire templates contain control information to the Data Provider Client in XML format. The Data Provider Client uses this information to create the HTML user page for entering data; also contributing developers must interpret this to generate correct import files. Templates are different for each survey; however, the structure of XML files containing template description are based on a standard XSD (see iform-sablon.xsd), thus allowing for every template to be logically interpreted and used regardless of implementation tools and languages. The basic principle is that only properties different from defaults are included in templates.

Data import into KSH-ELEKTRA is possible using XML files generated by other information systems. These files are further referred to as **data XML**. The Data Provide Client creates such data XML's when editing/saving a questionnaire. The handling of data XML's is controlled by questionnaire templates created for each type of survey.

The structure of data XML's follows a uniform logic (as shown in the attached file iform-adatxml.xsd); the actual structure of the file is different for each survey. The internal structure can be built by analyzing the questionnaire template. **Data XML files must use UTF-8 encoding.**

Questionnaire structure, fundamentals

Apart from graphical elements supporting manual fill-in, a questionnaire consists of an arbitrary number of chapters, each of which contains input fields and component structures (tables) and repeated items (table rows) of an arbitrary number. The latter can be nested, so the complexity of individual chapters is unlimited in theory – it is finite in practice, to keep the questionnaire manageable. The system handles tables only in the data XML; indeed, it groups related data, and in case they are repeated, constructs a row from each group of related data, arranging them in a table. With repetitions, it can be specified in the template whether the user can add new rows to those personalized for them initially, or to delete personalized rows.

A group of repeatable data cannot always be arranged in a table; however, the system allows entire chapters to be repeated in multiple instances. This design pattern is logically analogous to repeating table rows, but on a higher level. The behavior of chapter repetition can be controlled in a way identical to repetition of table rows.

The creation of structures in a data XML is controlled by the questionnaire template.

- 1) An instance of a <chapter> element is present for each chapter of the questionnaire instance.
- 2) Each input component and sequence number field (whether it is shown or hidden) is specified as a <data> element, which contains 2 elements:

- <identifier>: shows which actual component of the questionnaire template this item is derived from
- <value>: data value entered into this input field.

The <data> element can have the attribute “s”, which shows the source of the data item. For data prepared by KSH, its value is “P”, and it is to be preserved in the data XML. For data entered by the data provider, the “s” attribute must not be specified.

3) <data> elements are organized into tables and rows in the following cases.

- Repeatable panels: every instance creates a table row.
- Record pointer panels: components with identical record pointer values form a table row.
- Data merged from multiple panels: for a panel marked for merging, its child panels do not form separate tables; instead, the repetitions of child panels (or rows determined by record pointers) are combined into one big table, referring to the common parent.

The latter two are further described in Data XML Structure.

4) Creating tables as described above is also possible with nested structures.

Data XML files exported from questionnaire in the KSH-ELEKTRA system is a good basis for validation of data XML’s created externally. The file can contain, apart from data entered by the user, fields and data items loaded into the data XML by personalization, marked in the element with an attribute value s=“P”. These items cannot be modified by import; however, if you plan to run the validations/calculations defined in the questionnaire, expect that you may need these data as well.

<attachments> and <attachment> tags, available in the form technology built into KSH-ELEKTRA, are not actually used by the system.

However, to store error explanations, we apply the <justifiableErrors> tag after the <chapter> tags, along with its child elements, with the identifier, user-supplied content and other description data of each error explanation.

Important note: the fields used to identify the questionnaire are loaded into the data XML of each questionnaire instance as defined by KSH. These data must be reproduced in data XML’s generated by external systems so that the system can successfully import the questionnaire. These fields are present in two locations in the data XML:

1. The attributes identifying the template are specified in the <templateKeys> element at the beginning of the data XML. For each item, there is an <element> with a <name> and a <value> child element.

| Value of <name> | Description |
|------------------------------|--------------------------|
| mc01 | Survey number (OSAP) |
| mev | Year of data declaration |

For technical reasons, other data can be present as children of <templateKeys>. They need not be specified in external files to be uploaded; the system keeps their values stored on the server in any case.

Identifier fields in Preface (chapter 0), stored in <data> tags as normal input fields.

| Value of <name> | Description |
|------------------------------|--------------------|
|------------------------------|--------------------|

| | |
|--------|---------------------------------------------------------------------|
| MHO | Month of declaration |
| M003_G | Year of data declaration |
| M003 | Identifier of reporting unit |
| MV42 | Identifier of reporting subunit (mandatory only in certain surveys) |

Data XML files must be named according to the following convention.

- Without mutation (assuming mutation is 0):
 - adat_[OSAP code in 4 characters]_[year in 2 characters]_[unique client side identifier without an underscore].xml
- With mutation specified:
 - adat_[OSAP code in 4 characters]_[year in 2 characters]_[mutation]_[unique client side identifier without an underscore].xml

The KSH-ELEKTRA system does not impose restrictions on the client side identifier, only restrictions of the user's operating system apply

Apart from the general principles and specifications described above, the Developer's Guide also contains

- A questionnaire template file for each survey, named [OSAP code]_[year of declaration]_[template version].xml, or [OSAP code]_[year of declaration]_[mutation]_[template version].xml
- A technical PDF for each survey, visually displaying the structure of the questionnaire and the (database column) property for each input field. This information can also be extracted from the questionnaire template, as specified in the template XSD.

Proposals for data XML generation

- Always be careful to use UTF-8 encoding.
- Instead of generating the entire data XML from scratch, it is advisable to download a data XML prepared by KSH from a questionnaire in ELEKTRA, and to use it as a pattern.
- It is recommended to perform a schema validation on the generated data XML, using the file iform-adatxml.xsd.
- We recommend to test the data XML by importing it into a questionnaire opened in ELEKTRA, using the function "Import from data XML", and to verify the content on screen.

Attachments

- [Data XML Structure \(iFORM\)](#)
- [Data XML Schema \(iFORM\) – iform-adatxml.xsd](#)