An evaluation of Delta, a documentation tool

Rebecca Gatward, Office for National Statistics, UK.

1. Introduction

Since the 1980s CAI has become the main method of survey data collection. CAI software allowed researchers to meet the demand need for more complex survey questionnaires. The demand for a tool to document CAI questionnaires followed. CAI users throughout the survey process require a 'human readable' version of electronic questionnaires, for example, survey commissioners, researchers, methodologists, interviewers, data analysts and archivists.

The need for 'human readable' versions of Computer Assisted Interview (CAI) questionnaires is now well accepted. This issue described as the 'Global documentation problem' has been well discussed and debated at workshops, conferences, seminars and in papers. This paper will not revisit these arguments as these have been documented elsewhere. Bethlehem (2001), *'Survey Automation: report and workshop proceedings'* (2003)

Delta is the first documentation tool to be incorporated within the Blaise software. This paper will provide an early assessment of the Delta tool and will provide some recommendations for further development. The evaluation will be based on the set of function requirements that were the outcome of the User Requirement Survey which formed part of the TADEQ project. Delta developed from the TADEQ tool.

CAI has many advantages over paper questionnaires except to date, when it comes to documentation. During the early stages of CAI software development, developers of the CAI software packages quite rightly did not rank documentation tools as high priority. They concentrated their efforts on producing systems that would securely collect accurate data. Once this was achieved developers then turned their attention to meeting the demands of their growing numbers of users by improving the functionality of the packages. Producing a 'human readable' version of a CAI questionnaire is one function that has, until recently, not been tackled by the developers of CAI software. This forced survey organisations to tackle the problem themselves by developing their own 'in-house' documentation tools. Examples include the tool developed by the Office for National Statistics (ONS), which could automatically produce documentation from the Blaise Metadata files, and the US Census Bureau funded IDOC (Instrument Document) for CASES.

What occurred was a situation where organisations were working independently to solve the documentation problem themselves. The TADEQ project aimed to coordinate these efforts and provide a joint approach to finding a solution to this problem.

1.1 The TADEQ project

The TADEQ project (Tool for the Analysis and Documentation of Electronic Questionnaires) is a Fourth Framework Project of the EU. The project pooled the efforts of National Statistical Institutions from four European countries (Netherlands, United Kingdom, Portugal and Finland) and the Max Planck Institute in Germany[1]. In summary the aim of this project was to find out what Computer Assisted Interviewing users required from a questionnaire documentation tool (via a User Requirement Survey) and to develop a first prototype tool to meet these needs.

The outcome of the project was the TADEQ prototype which is capable of documenting and analysing the contents and structure of electronic questionnaires. TADEQ is designed to be a neutral tool that can be used to process any electronic questionnaire capable of being exported

[1] During the course of the project people involved at the Max Planck Institute moved to the Institute of Computer Graphics at the University of Vienna.

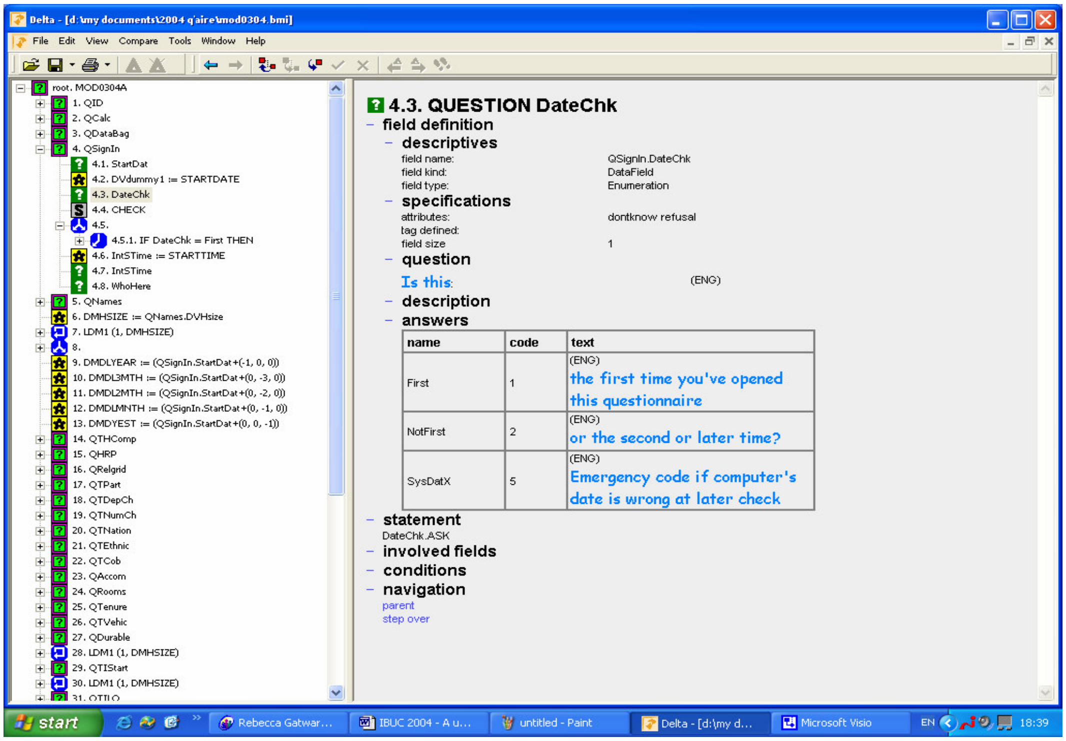
to XML. More details about the TADEQ prototype can be found in the 'Using the TADEQ prototype' (Bethlehem, Hundepool, Van de Wetering, 2000). There are overviews in earlier IBUC papers including (Bethlehem, J.G. and Hundepool, A.J. 2001).

1.2 Introduction to Delta

The first version of Delta was released as part of the Blaise Component Pack in June 2003. Delta is an acronym of Documentation tool, using ELements of TAdeq. As this name suggests, the Blaise developers at Statistics Netherlands have used the TADEQ prototype as a basis from which to build the Delta documentation tool. Their goal (as stated in the documentation that accompanied the Delta release) was to 'offer improved functionality of TADEQ, built upon Blaise components'. Other documentation tools have been developed by organisations using Blaise. However, Delta is the first such tool to form part of the Blaise software.

Blaise developers at Statistics Netherlands have taken some of the functionality of TADEQ and incorporated it into Blaise. As stated in the Delta on-line help, the two main elements of TADEQ that are not yet in Delta are the graph module which displays a questionnaire as a flow chart and the analysis options. A tool to compare two versions of a questionnaire is included in Delta.

The screen layout in Delta is almost exactly the same as TADEQ, with the tree view (displayed on the left of the screen) providing a means to navigate around the questionnaire. Detailed information about the questionnaire content appears on the right hand side of the screen. The format of the information and level of detail is controlled by stylesheets (written in XML). An example screen using the default stylesheet is provided below.



root. MOD0304A
1. QID
2. QCalc
3. QDataBag
4. QSignIn
4.1. StartDat
4.2. DVdummy1 := STARTDATE
4.3. DateChk
4.4. CHECK
4.5.
4.5.1. IF DateChk = First THEN
4.6. IntSTime := STARTTIME
4.7. IntSTime
4.8. WhoHere
5. QNames
6. DMHSIZE := QNames.DVHsize
7. LDM1 (1, DMHSIZE)
8.
9. DMDLYEAR := (QSignIn.StartDat+(-1, 0, 0))
10. DMDL3MTH := (QSignIn.StartDat+(0, -3, 0))
11. DMDL2MTH := (QSignIn.StartDat+(0, -2, 0))
12. DMDLMNTH := (QSignIn.StartDat+(0, -1, 0))
13. DMDYEST := (QSignIn.StartDat+(0, 0, -1))
14. QTHComp
15. QHRP
16. QRelgrid
17. QTPart
18. QTDepCh
19. QTNumCh
20. QTNation
21. QTEthnic
22. QTCob
23. QAccom
24. QRooms
25. QTenure
26. QTVehic
27. QDurable
28. LDM1 (1, DMHSIZE)
29. QTIStart
30. LDM1 (1, DMHSIZE)
31. QTILO

4.3. QUESTION DateChk

– field definition

– descriptives

field name:	QSignIn.DateChk
field kind:	DataField
field type:	Enumeration

– specifications

attributes:	dontknow refusal
tag defined:	
field size	1

– question

Is this: (ENG)

– description

– answers

name	code	text
First	1	(ENG) the first time you've opened this questionnaire
NotFirst	2	(ENG) or the second or later time?
SysDatX	5	(ENG) Emergency code if computer's date is wrong at later check

– statement

DateChk.ASK

– involved fields

– conditions

– navigation

parent
step over

2. Evaluation of Delta

2.1. Criteria for evaluating Delta

One of the aims of the TADEQ project was to establish a set of user requirements for a documentation tool. A survey was carried out amongst users of CAI software to collect data about what they required from a documentation tool. The information collected formed a set of function requirements for TADEQ.

A summary of the TADEQ user requirements are provide below, a detailed commentary of these requirements can be found in Kelly and Kuusela (2000).

The TADEQ project found that users required a documentation tool to:

- automatically generate accurate documentation that requires little or no manual editing,
- produce comprehensive documentation which provides all the information a user requires,
- meet the needs of different users by providing a flexible tool which allows the information displayed or printed to be changed,
- generate documentation in paper and electronic form,
- produce detailed information about questions and the logical structure of the questionnaire.

As the survey process has not changed radically since the TADEQ user requirement survey was carried out in 1998, we can assume that the user requirements for a documentation tool still remain the same. The user requirements, listed above, will therefore be the criteria by which we will assess Delta. The next few sections will take each one in turn assessing to what extent Delta meets these requirements.

2.1.1. Accurate documentation

It is essential that a documentation tool automatically produces output that requires little or no manual editing to make it fit for purpose. Any manual intervention risks introducing inaccuracies, even if the person carrying out the amendments knows the questionnaire very well. Any manual editing of documentation makes it difficult to guarantee the documentation still replicates exactly what is happening in the CAI questionnaire. Manually crafting documentation to meet user needs can also take a long time, resulting in some of the efficiencies gained by using CAI rather than PAPI are then being lost on the documentation process.

Delta partly meets this criterion. The information displayed in Delta documentation does accurately reflect the information gathered in the Blaise questionnaire. This is because Delta extracts meta-information, data definition and rules that are held in the Meta information file (.bmi) along with screen layout information stored in the DEP mode files (.bdm) and translates it into Delta output.

Generating documentation using Delta is automatic and very straightforward, the user simply selects the questionnaire .bmi file and Delta returns the documentation on screen.

Delta also allows the user to manually edit what appears in the documentation. There are two main reasons why the user may wish to do this. The first is to tailor what is displayed on screen to meet user needs, the second to edit the documentation before it is printed. Manual editing of Delta documentation will be dealt with in more detail when paper and electronic versions are considered later in the paper.

2.1.2. Comprehensiveness

The second user requirement to be considered is all the information required by a user should be available in the documentation (even if the user chooses not to display it on screen).

Different users require different types of information at varying levels of detail. For example, one user may only need to view variable names and question text while another user may require detailed routing for each variable.

A table of all the information that the TADEQ user requirements survey found necessary is included as an appendix to this paper (Appendix A); the table provides some detail about whether or how Delta provides each item of information.

In summary, Delta displays most of the information users require, although some items are not shown in the most accessible format for the user.

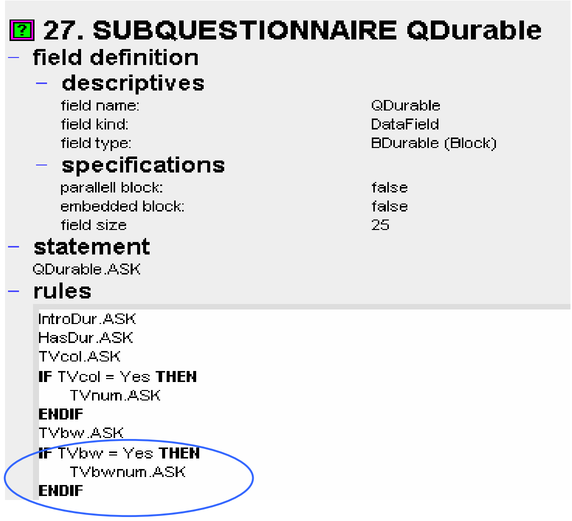
The following items are not currently provided by Delta but are required.

- A ‘natural language’ description of the questionnaire filename.
- Variable attributes.
- References to external files
- Information about the type of coding frame or type of search (e.g. Trigram)
- Details of external programs or DLLs referred to in the questionnaire.
- Text substitutions (fills) are not labelled or translated.
- ‘SET OF’ details to identify multi-coded variables.
- Natural language explanation of routing logic.
- Variable index (or option to list all variables).
- Information imported from external programs.
- Contents of an external coding file.
- LAYOUT commands displayed at block or field level.

I will take some of the more important requirements in more detail, starting with the way routing is dealt with by the Delta tool.

Routing

In Delta routing is displayed at three different points, at block (or sub questionnaire) level, in the tree view and at the Field level. In the example below the routing to the variable called ‘TVbwnum’ is highlighted at these three points.



27. SUBQUESTIONNAIRE QDurable

− **field definition**

− **descriptives**

field name:	QDurable
field kind:	DataField
field type:	BDurable (Block)

− **specifications**

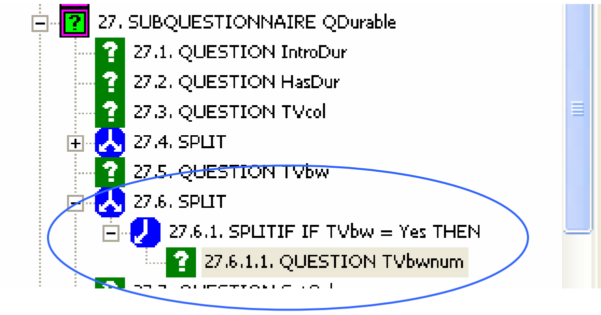
parallell block:	false
embedded block:	false
field size	25

− **statement**

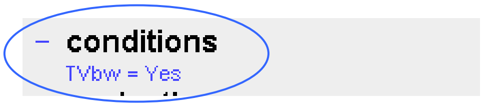
QDurable.ASK

− **rules**

```
IntroDur.ASK
HasDur.ASK
TVcol.ASK
IF TVcol = Yes THEN
    TVnum.ASK
ENDIF
TVbw.ASK
IF TVbw = Yes THEN
    TVbwnum.ASK
ENDIF
```



27. SUBQUESTIONNAIRE QDurable
27.1. QUESTION IntroDur
27.2. QUESTION HasDur
27.3. QUESTION TVcol
27.4. SPLIT
27.5. QUESTION TVbw
27.6. SPLIT
27.6.1. SPLITIF IF TVbw = Yes THEN
27.6.1.1. QUESTION TVbwnum



– conditions
TVbw = Yes

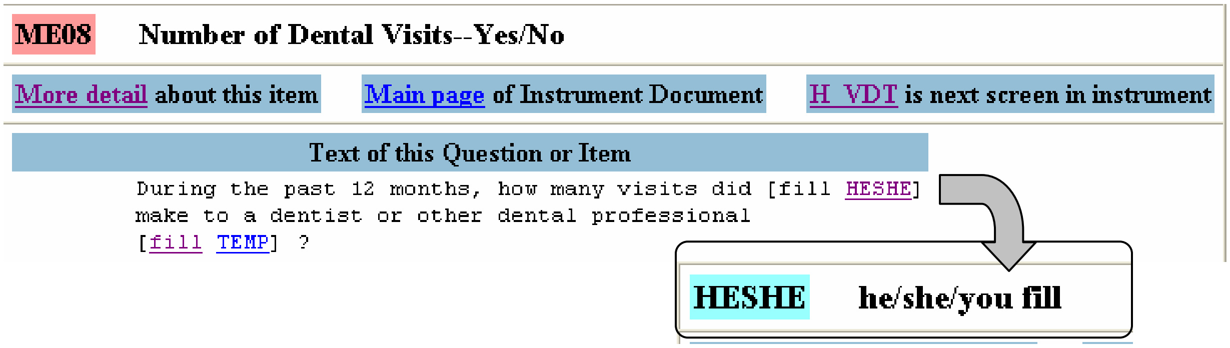
The routing is displayed in the same format as it is written in Blaise code, i.e. as IF and ELSE statements. This format may be useful for some users but not others. For example, a survey commissioner may ask for documentation to include the full routing to each variable whereas an interviewer may find routing in the 'Go to' format (as it appears in paper questionnaires) easier to follow.

Users need more flexibility around changing the format and complexity of the routing displayed. They need the option to display routing at varying levels of complexity, for example, routing between blocks (sub questionnaires) or to each field. The routing also currently uses value names: these are not always meaningful and make deciphering routing a lengthy process. It would be helpful to have the option to choose whether the value labels or value names are shown in the routing.

Questionnaire developers attempt to help users interpret routing by annotating the rules sections with natural language descriptions of the routing logic. Unfortunately this information is not currently included in the Delta output (see Appendix A).

Text substitutions

A second information requirements to be looked at in more detail is question text, specifically text substitutions. Although the format of the question text in the documentation tool is the same as it appears in the Data Entry Program, text substitutions appear as they do in the Blaise syntax. Text fills are not translated or labelled. Other documentation tools however have managed to do this. The IDOC (Instrument Document) tool (developed on behalf of the US Census Bureau) deals with fills by labelling them in the question text e.g. [fill 'variable name'] this is highlighted with links to the 'variable name' where the fill was originally defined.



ME08 **Number of Dental Visits--Yes/No**

More detail about this item | Main page of Instrument Document | H_VDT is next screen in instrument

Text of this Question or Item

```
During the past 12 months, how many visits did [fill HESHE]
make to a dentist or other dental professional
[fill TEMP] ?
```

HESHE **he/she/you fill**

This solution is not ideal because the user still has to go elsewhere to find the contents of the text fill. It is essential for Delta to provide the user with some indication of what text substitution means. Other ways of dealing with text fills are described in Kelly and Kuusela (2000).

Imported information

The TADEQ requirement survey also found that users would like to be able to import other information about a survey into the documentation tool. For example, details about the survey methodology (sample size, fieldwork dates) or history of changes to a questionnaire. It is more convenient to keep all this metadata in one document and would make it a document that is fit to be sent to data users. This is not currently possible in Delta. A solution could be to have the option of including an ‘information’ block within the Blaise questionnaire where any additional information about a survey could be stored and extracted by Delta.

Other information that would be helpful is a summary of the settings applied to the questionnaire during its preparation or the interview. For example, the name of the Modelib file and details from within the Modelib such as whether the audit trail facility was activated.

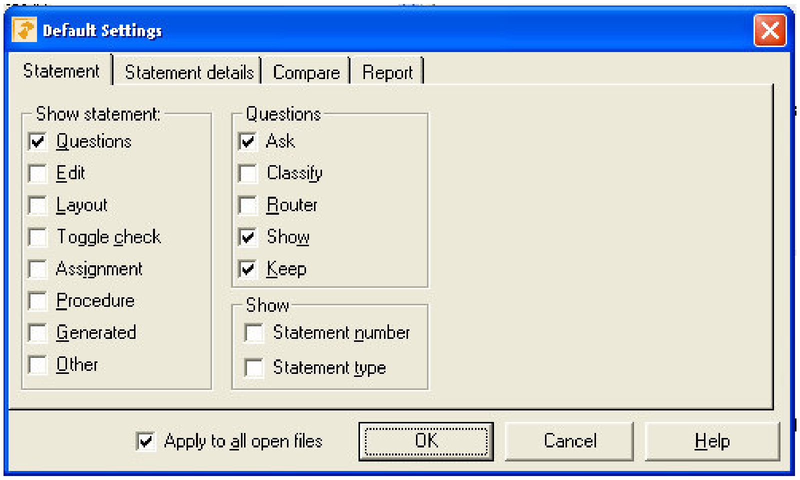
Currently any text which is specified at an Auxfield (typically an introduction to a new section of questions) is not displayed in Delta. Although this is not strictly question text, it is still important that this detail is available to users.

2.1.3. Flexible

As mentioned already, people who are involved at different points in the survey process all require different information from the questionnaire documentation. A documentation tool needs to be capable of producing the documentation, both in electronically and on paper, to meet these differing needs.

The tool needs to allow the user to be flexible in terms of language i.e. having the option to display natural language descriptions for variables and the variable names and also allow the user to choose the level of detail shown.

Delta does provide this flexibility. It is possible to specify which details are shown on screen or printed in the paper version. The user can choose what information is shown in the structure tree by changing the ‘default settings’. The screen shot below illustrates the options available.



Default Settings

Statement | Statement details | Compare | Report

Show statement:
- ☑ Questions
- □ Edit
- □ Layout
- □ Toggle check
- □ Assignment
- □ Procedure
- □ Generated
- □ Other

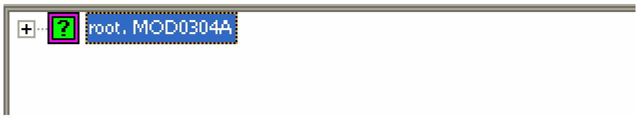
Questions
- ☑ Ask
- □ Classify
- □ Router
- ☑ Show
- ☑ Keep

Show
- □ Statement number
- □ Statement type

☑ Apply to all open files

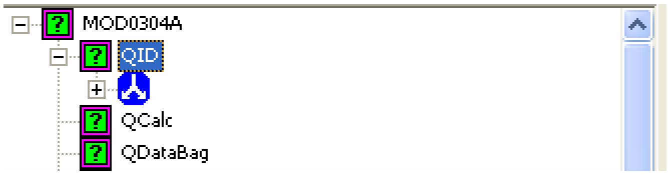
An additional user requirement was the ability to show the global structure of the questionnaire and to also have the means to focus on the detail of parts of the questionnaire. The following example illustrates that this is possible in Delta.

1.



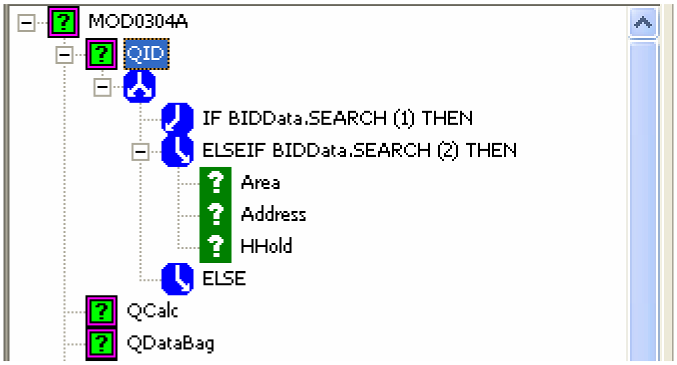
```
[+] root. MOD0304A
```

2.



```
[-] MOD0304A
    [-] QID
        [+]
    QCalc
    QDataBag
```

3.



```
[-] MOD0304A
    [-] QID
        [-]
            IF BIDData.SEARCH (1) THEN
            [-] ELSEIF BIDData.SEARCH (2) THEN
                Area
                Address
                HHold
            ELSE
    QCalc
    QDataBag
```

A further user requirement is that it should be possible to delete some sections of questions from the documentation (for e.g. questions that survey commissioners do not need to see like administrative details). Currently this would not be possible in Delta, unless the sections were simply removed from a paper version or an alternative questionnaire was prepared which excluded the relevant questions.

2.1.4. Paper and electronic format

A primary requirement for TADEQ was to produce well laid out, good documentation in both an electronic and paper format. The documentation should be capable of producing documentation in either format which meets all user needs. Some
components of a questionnaire are important to one type of user but less so to another.

- Interviewers are mostly interested in question text, on screen instructions, routing and on screen help instructions.
- Survey commissioners need to see the question wording, routing, checks and computations.
- Data users require question routing and computations.

Paper version

Even in the age of the 'paperless office', paper versions of CAI questionnaires are still required by users of CAI software. People working throughout the survey process require paper versions of questionnaires for a variety of purposes. Some use the paper version as an easy reference document, such as field staff whilst dealing with interviewer queries or researchers whilst analysing data. It is sometimes easier and quicker to reach for a paper copy than to open up an electronic version. Many people also find it quicker to read and navigate a paper copy than an electronic version.

Currently the paper documentation offered by Delta is simply a print out of the electronic version. This format is not ideal for a paper version. It is possible to tailor what is printed to meet user's needs. Information (statement details) appear on the right hand side of the screen, and stylesheets are used to control how this is displayed. Currently however only one default stylesheet is provided with Delta so it is down to the user to amend the stylesheet. Amending the stylesheet, which is written in XML, is not a straightforward task especially for a 'non-programmer'. At least it is possible to save the stylesheets which means that stylesheets only have to developed once to meet different user requirements or to a 'housestyle'. This task could be simplified for the non programmer if a stylesheet editor was provided with Delta (like the one provided with Blaise IS).

Printing a paper copy of the Delta documentation is also problematic. It takes a long time to print (the on-screen estimate was 60 minutes for a surveys of average length and complexity) and frequently crashes the PC before the whole document has printed.

The Delta tool allows the user to add some formatting to a paper version, such as page numbers, headers and footers. This makes it easier for the user to find their way around what can be a very long document. Other options are print preview, the option to print just the tree view, the statement details or all details and also to print more than one page per sheet. Delta does not give the user the option to add an index or contents page or to print more than one variable on one each sheet. Having an option available to produce an index based on variables names is particularly important.

Exporting the output to a package that offers more formatting options is possible but would require saving the document as a HTML file (the only option available in Delta) then resaving this version as another type of file. This process is time consuming and additional time would then have to be spent reformatting the text.

The advantage of saving the Delta output as a HTML file is that the functionality of the document remains within the document, for example the ability to expand the branches. However saving the document is a very long process, the onscreen estimate for gathering and saving all details in a questionnaire of average length and complexity was 53 minutes.

The addition of an index of variables would be helpful. This could be used in both the paper and electronic versions. A possible format is a list all variable names alongside a brief description of the variable, with links that allow the user to jump to any variable within the documentation.

Many websites now offer the option of printing a 'printer friendly' option that is formatted specifically to be printed rather than viewed on screen. Delta would benefit from a similar feature.

Electronic format

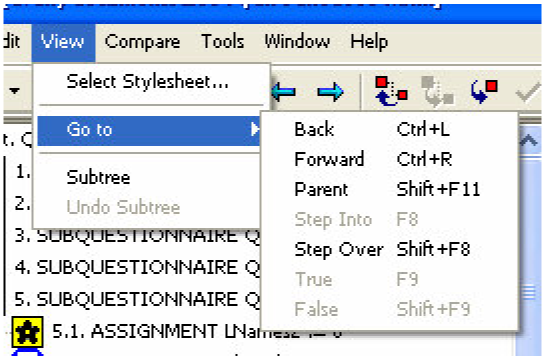
Some users who took part in the TADEQ requirement survey did use electronic versions of documentation (although all still used a paper version as well). The advantages of electronic documentation are interactive use, easier navigation around the questionnaire, search and help facilities available on-line and option to use standard windows facilities, for example, being able to copy text into a different package.

Interactive use

As described under the 'flexible' criterion above, Delta allows the user to show and hide information as required.

Navigation

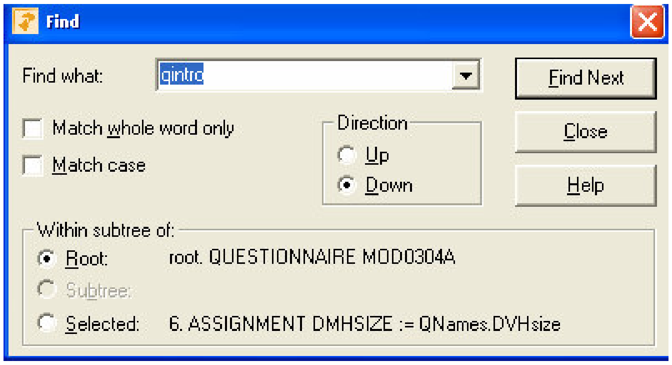
Navigation is via the tree view on the left hand side of the screen. This system of navigation works well with small questionnaires but it is easy to get lost in the detail when looking at a large complex survey. Different symbols are used to identify the various statements (such as a white question mark with a dark green background representing a question), this is helpful. The user can also find their way around the documentation using the 'Go to' options (shown below) or the hyperlinks within the document.



Navigation could be improved by adding links to variable names, this would allow the user to jump to variables mentioned in the condition or rules sections rather than using the search facility. A further improvement would be the addition of an ‘expand all’ and ‘collapse all’ option to the ‘view’ menu.

Search and help facilities

The search (or ‘Find’) option in Delta allows the user to search for any text within the documentation, either within the whole document (from the ‘Root’) or within selected sections. Each occurrence of that text string is highlighted on screen.



Find what: qintro

☐ Match whole word only

☐ Match case

Direction: ○ Up ● Down

Within subtree of:

● Root: root. QUESTIONNAIRE MOD0304A

○ Subtree:

○ Selected: 6. ASSIGNMENT DMHSIZE := QNames.DVHsize

The Help facility provides more detail about all aspects of the Delta tool.

Standard window utilities

It is only possible to save the statement details as an HTML file and the tree as a JPEG. It is not possible to export to other packages. Exporting to WORD would be useful.

2.1.5. Ability to produce detailed information on the logical structure of the questionnaire

The Delta tool only incorporates some of the functionality of the TADEQ prototype. The functions that are currently missing from Delta are the analysis and graphical tools; it is these elements that would meet this final user requirement.

Graphical tool

Many of the users who took part in the TADEQ user survey stated that a flow diagram would help them and other users gain an overall view of the flow of a questionnaire. For example, a flow diagram would be useful for interviewers to gain a picture of how respondents with differing characteristics would be routed through a questionnaire. The flow diagram is also an ideal format to display ‘go to’ routing.

Users should also be able to use the flow chart to drill down to sections of the questionnaire they are particularly interested in, for example, from the overall structure of questionnaire down to the routing within a block and then the logic behind the derivation of a computed variable. These functions are available in TADEQ.

Jabine (1985) advocates flow charts as a useful tool that can aid the questionnaire development process, both in the early and later stages of design and also for training interviewers.

Analysis tool

The TADEQ user requirement survey did collect some information from users about what they needed from an analysis tool. These are detailed in Kelly and Kuusela (2000).

The following analysis functions were available in the TADEQ prototype.

- Calculation of the number of paths through the questionnaire.
- average length of interview, calculated using estimated time taken to answer each question.
- Frequency of the different elements of a questionnaire for e.g. number of checks, open questions etc.

Compare tool

Alongside the documentation element a comparison function has been included in Delta. The compare tool compares two versions of a questionnaire finding and documenting the differences between them. This is a time saving tool, although useful enhancements would be the addition of an option to select sections of the questionnaire to compare rather than whole questionnaire and an option to select changes that can be ignored, such as additional toggles around text and spaces.

3. Summary and recommendations for future development

The first version of Delta partially meets some of the TADEQ user requirements. In its current form Delta:

- automatically generates documentation very quickly,
- provides some of the information users need,
- is flexible – in that the user can choose what and how information is displayed,
- provides documentation in an adequate electronic format,
- provides a tool to compare two questionnaires.

Aspects that require particular attention for development are:

- use of natural language, e.g. for labels, for all Blaise elements, including computations,
- displaying routing in a more meaningful way,
- providing a 'printer friendly' format option,
- providing a graphical or analysis tool that meets user requirements.

In addition to the recommendations detailed under each evaluation criterion, the following recommendations that were not explicitly found to be user requirements by the TADEQ project but would be very useful enhancements to the Delta tool.

Use Delta to analyse audit trail data

Delta could be developed to be a tool to analyse and summarise audit trail data. The analysis component of the TADEQ prototype calculates an average interview length based on estimated length of time taken to ask each question. Now that actual timings can be calculated using audit trail data, Delta could provide accurate timings rather than estimates. The tool could produce actual interview length and average block timings. These timings could be provided for the whole sample or by respondent type. Users could be given the option of selecting the type of respondent, for example, females aged 60 years and over, Delta would list the questions which could be asked of that sub sample and the average interview length.

Alongside the work required from the Blaise developers, Blaise users must also play their part in improving documentation. Delta can only extract what is in the Blaise questionnaire. Questionnaire developers need to be encouraged to add descriptions and annotations in natural language; this is in line with the general push for more accurate and thorough metadata throughout the world. Blaise may need to be enhanced to permit labels in all elements.

4. References

Bethlehem, J.G., Hundepool, A.J. and Van de Wetering (2000) A. *Using the TADEQ prototype. TADEQ Working paper 6*, Statistics Netherlands, Voorburg, The Netherlands

Bethlehem, J.G. and Hundepool, A.J.(2001) *TADEQ: A Tool for the Analysis and Documentation of Electronic Questionnaires* Pre-proceedings of the ETK-NTTS 2001 Conference, Volume 1, pp. 133-143

Bethlehem, J.G. (2001) *Documentation of electronic questionnaires*, Statistics Netherlands, Voorburg, The Netherlands 2001

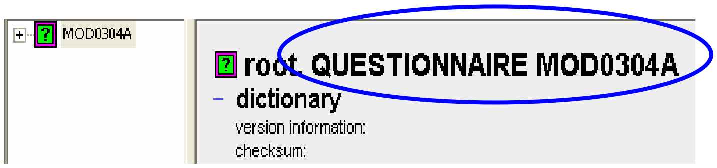
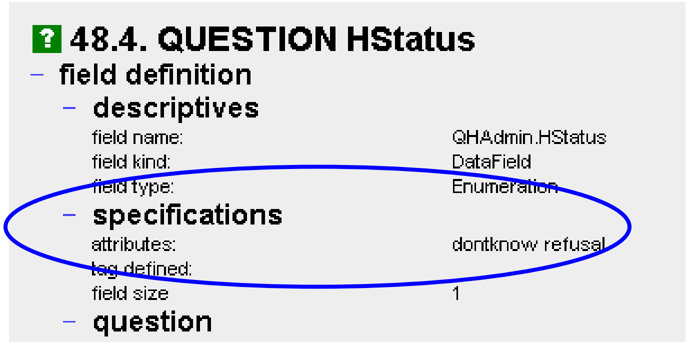
Bethlehem, J.G. and Hundepool, A.J. (2001) *Automated Questionnaire Documentation. Research Paper 0124*, Statistics Netherlands, Methods and Informatics Department. Also presented at the 7th International Blaise Users Meeting, Washington D.C., 11 September 2001

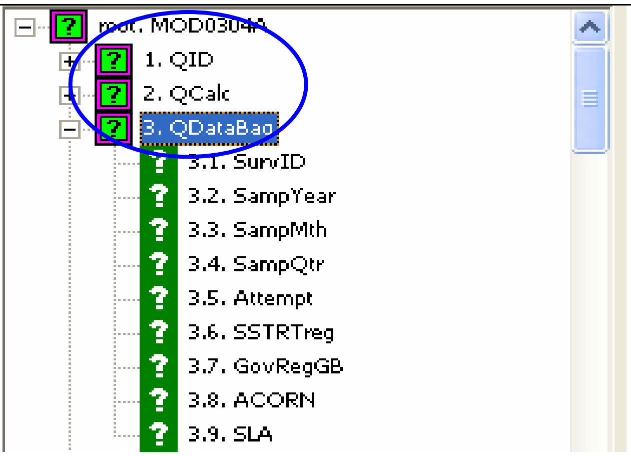
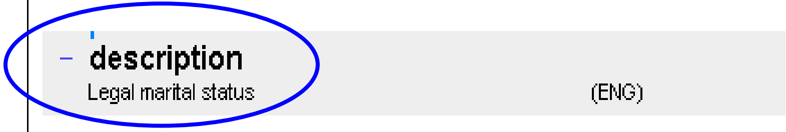
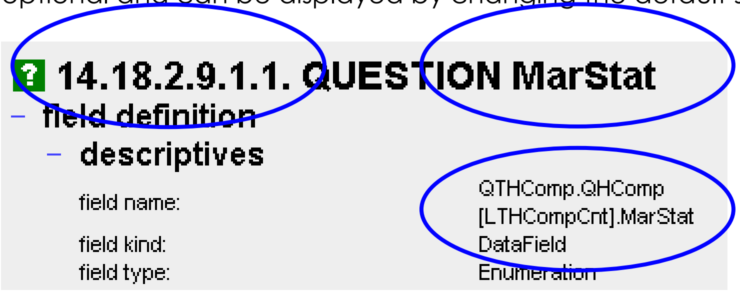
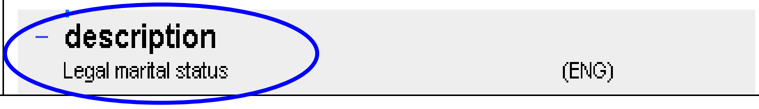
Jabine, T. P. (1985) *Flowcharts: A tool for developing and understanding survey questionnaires*, Journal of Official Statistics 1, pp. 189-207

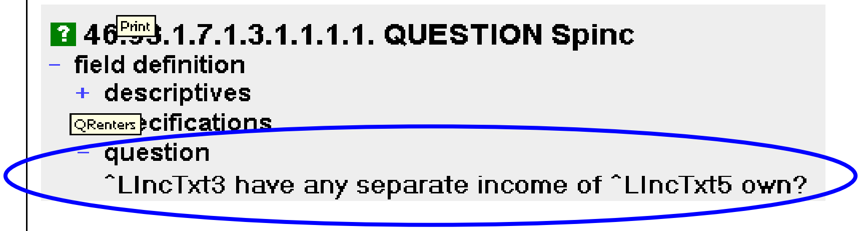
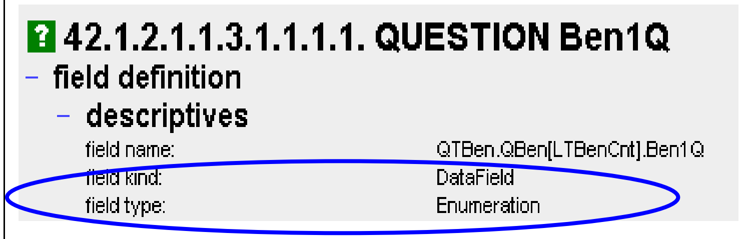
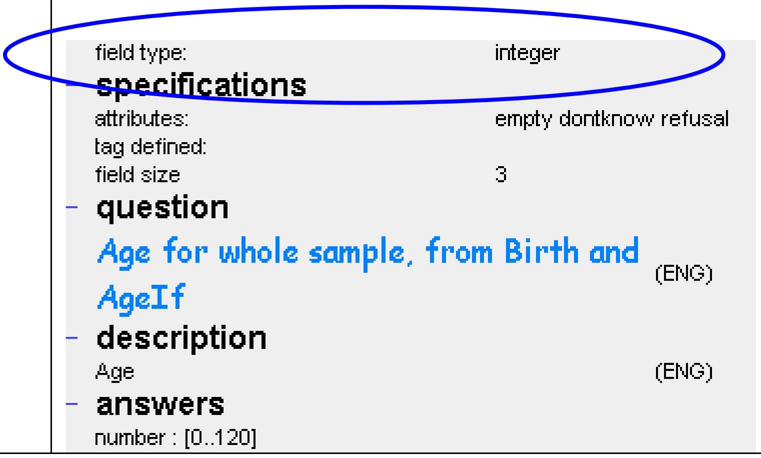
Kelly, M. and Kuusela, V. (2000) *User requirements TADEQ Working paper 1*, Statistics Netherlands, Voorburg, The Netherlands

'Survey Automation: report and workshop proceedings' Committee on National Statistics (CNSTAT) 2003

Appendix A - Information users require from a documentation tool

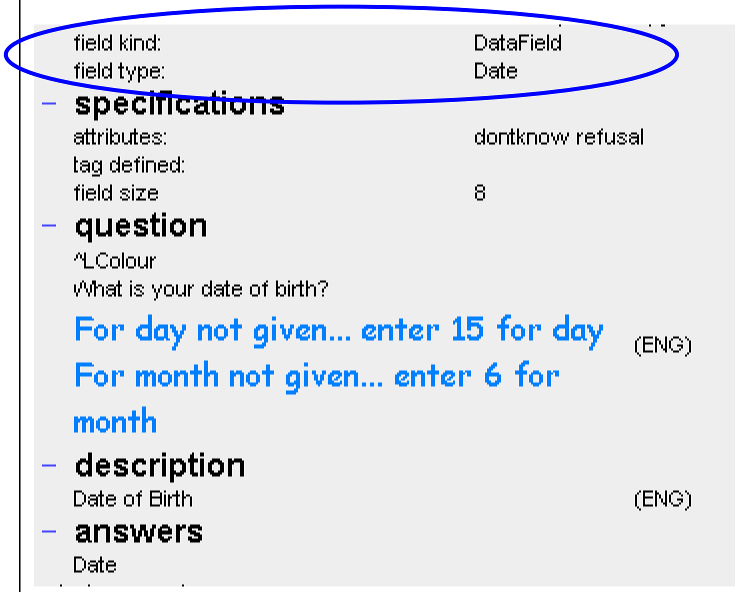
Information required	What is available in Delta	
Questionnaire file name and description	Filename is provided at the highest level of the tree. File description is not shown even when this information is included in the Blaise syntax. For example: • the Blaise syntax `DATAMODEL MOD0304A "BDSS code for HOUSEHOLD surveys"` • how this appears in Delta output  MOD0304A root. QUESTIONNAIRE MOD0304A – dictionary version information: checksum:	✓ ✗
Global and variable attributes	Global attributes are not displayed or available in the documentation produced by Delta. Variable attributes are displayed, for example:  48.4. QUESTION HStatus – field definition – descriptives field name: QHAdmin.HStatus field kind: DataField field type: Enumeration – specifications attributes: dontknow refusal tag defined: field size 1 – question .	✗ ✓
External files	References to external files are not displayed in the Delta output. Information about the type of coding frame or search is also not included.	✗ ✗
External programs/DLLs	Details of any external programs or DLLs referred to in the Blaise questionnaire are not displayed in Delta.	✗
Questionnaire sections	Questionnaire sections are listed in the tree view and can be used to navigate around the questionnaire. An example:	✓

	 root. MOD0304A 1. QID 2. QCalc 3. QDataBag 3.1. SurvID 3.2. SampYear 3.3. SampMth 3.4. SampQtr 3.5. Attempt 3.6. SSTRTreg 3.7. GovRegGB 3.8. ACORN 3.9. SLA Delta allows the user the option to display individual sections or the whole questionnaire on the screen.	✓
Structure diagram	The structure diagram (tree) in Delta can be expanded or collapsed to display an overview or to focus on sections of questionnaire in more detail. This diagram can also be printed if required.	✓
Natural language descriptions	The variable description (a short meaningful description of the variable) is one of the statement details that are displayed on the left hand side of the screen in Delta. As long as that information is added in the Blaise questionnaire by the questionnaire developer. e.g. The description for a variable named MarStat is specified in the Blaise code as 'legal marital status' . This description is displayed in Delta.  – description Legal marital status (ENG)	✓
Information about variables • **Full name** • **Short name** • **Variable numbers** • **Variable description**	All the information about variables listed as a 'user requirements' are displayed in Delta documentation. The variable numbers are optional and can be displayed by changing the default settings.  14.18.2.9.1.1. QUESTION MarStat – field definition – descriptives field name: QTHComp.QHComp [LTHCompCnt].MarStat field kind: DataField field type: Enumeration  – description Legal marital status (ENG)	✓

Question text	Question text is displayed as it is written in the Blaise syntax. However, text substitutions (fills) are not labelled or translated. For e.g.  46.33.1.7.1.3.1.1.1.1. QUESTION Spinc – field definition + descriptives ecifications – question ^LIncTxt3 have any separate income of ^LIncTxt5 own?	✓ ✗
Answer types	Information about answer types are displayed on screen. Some examples: • **Enumerated field** Although enumerated types are identified, although there is nothing displayed to indicate whether the variable is a multi-code. In the example below it is only possible to identify this because 'Code all that apply' instruction is included in the question text and displayed in infopane.  42.1.2.1.1.3.1.1.1.1. QUESTION Ben1Q – field definition – descriptives field name: QTBen.QBen[LTBenCnt].Ben1Q field kind: DataField field type: Enumeration The answer categories are also shown along with the value names and labels. • **String** String variables are labelled 'Integer' rather than string.  field type: integer specifications attributes: empty dontknow refusal tag defined: field size 3 – question Age for whole sample, from Birth and AgeIf (ENG) – description Age (ENG) – answers number : [0..120]	✓

<table>
<tr>
<td></td>
<td>

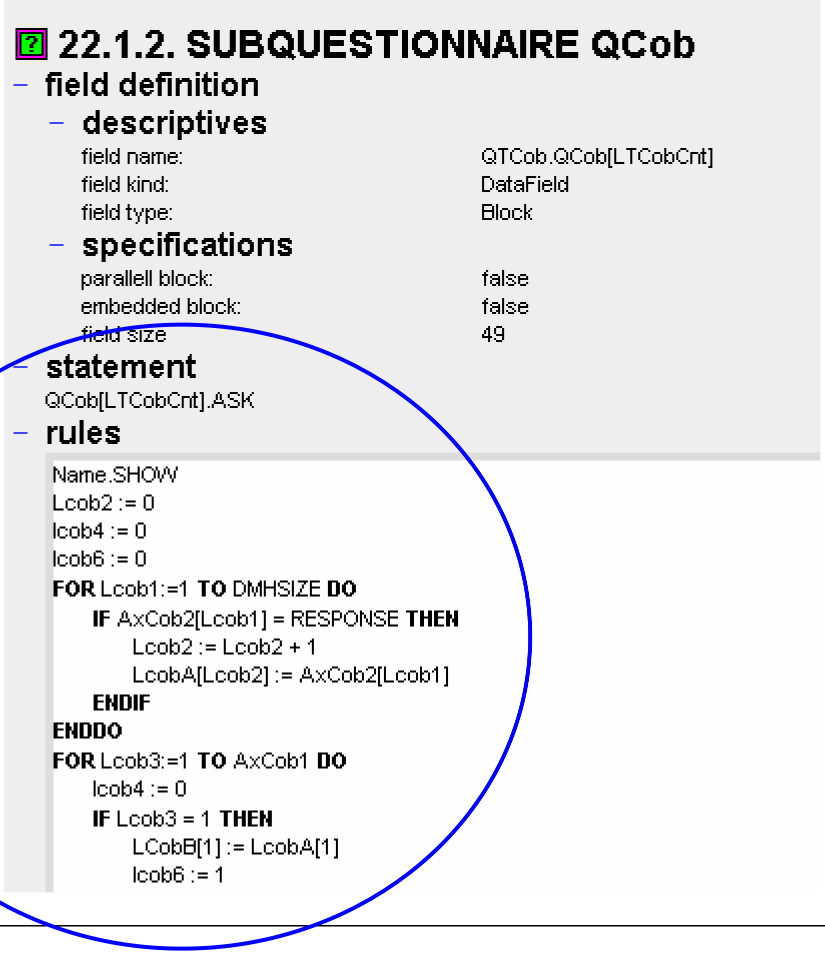
- **Datetype**



field kind: DataField
field type: Date
– **specifications**
attributes: dontknow refusal
tag defined:
field size 8
– **question**
^LColour
What is your date of birth?
For day not given... enter 15 for day (ENG)
For month not given... enter 6 for month
– **description**
Date of Birth (ENG)
– **answers**
Date

</td>
<td></td>
</tr>
<tr>
<td><b>Routing</b></td>
<td>

In Delta routing is displayed at three levels. The entire RULES section for an include file or block is displayed at the block level, they appear in the same format as they do in the Blaise syntax, for example:

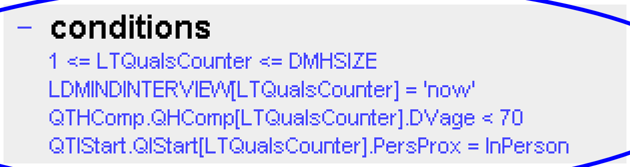


22.1.2. SUBQUESTIONNAIRE QCob
– **field definition**
– **descriptives**
field name: QTCob.QCob[LTCobCnt]
field kind: DataField
field type: Block
– **specifications**
parallell block: false
embedded block: false
field size 49
– **statement**
QCob[LTCobCnt].ASK
– **rules**

```
Name.SHOW
Lcob2 := 0
lcob4 := 0
lcob6 := 0
FOR Lcob1:=1 TO DMHSIZE DO
    IF AxCob2[Lcob1] = RESPONSE THEN
        Lcob2 := Lcob2 + 1
        LcobA[Lcob2] := AxCob2[Lcob1]
    ENDIF
ENDDO
FOR Lcob3:=1 TO AxCob1 DO
    lcob4 := 0
    IF Lcob3 = 1 THEN
        LCobB[1] := LcobA[1]
        lcob6 := 1
```

</td>
<td>✓</td>
</tr>
</table>

Routing to variables within the block are
Displayed at the variable level under the heading conditions.



```
– conditions
    1 <= LTQualsCounter <= DMHSIZE
    LDMINDINTERVIEW[LTQualsCounter] = 'now'
    QTHComp.QHComp[LTQualsCounter].DVage < 70
    QTIStart.QIStart[LTQualsCounter].PersProx = InPerson
```

It is possible to specify if the rules or conditions section are displayed.

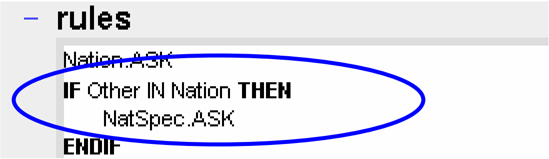
Information which has been added by the questionnaire developer that may help others understand the routing logic is not displayed in Delta.

In the example below the developer has annotated the rules section.
This information does not appear in the rule section in Delta.

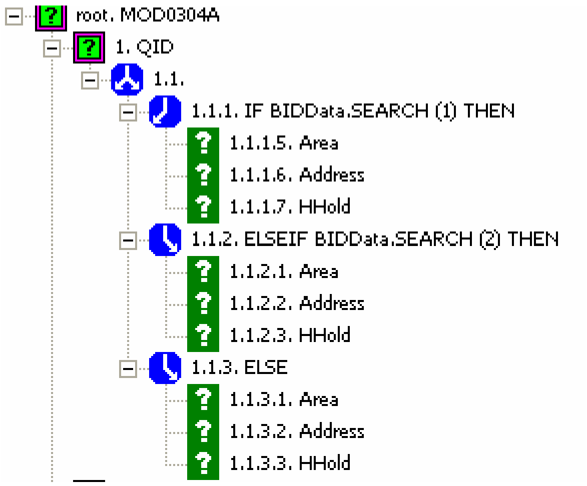
✗

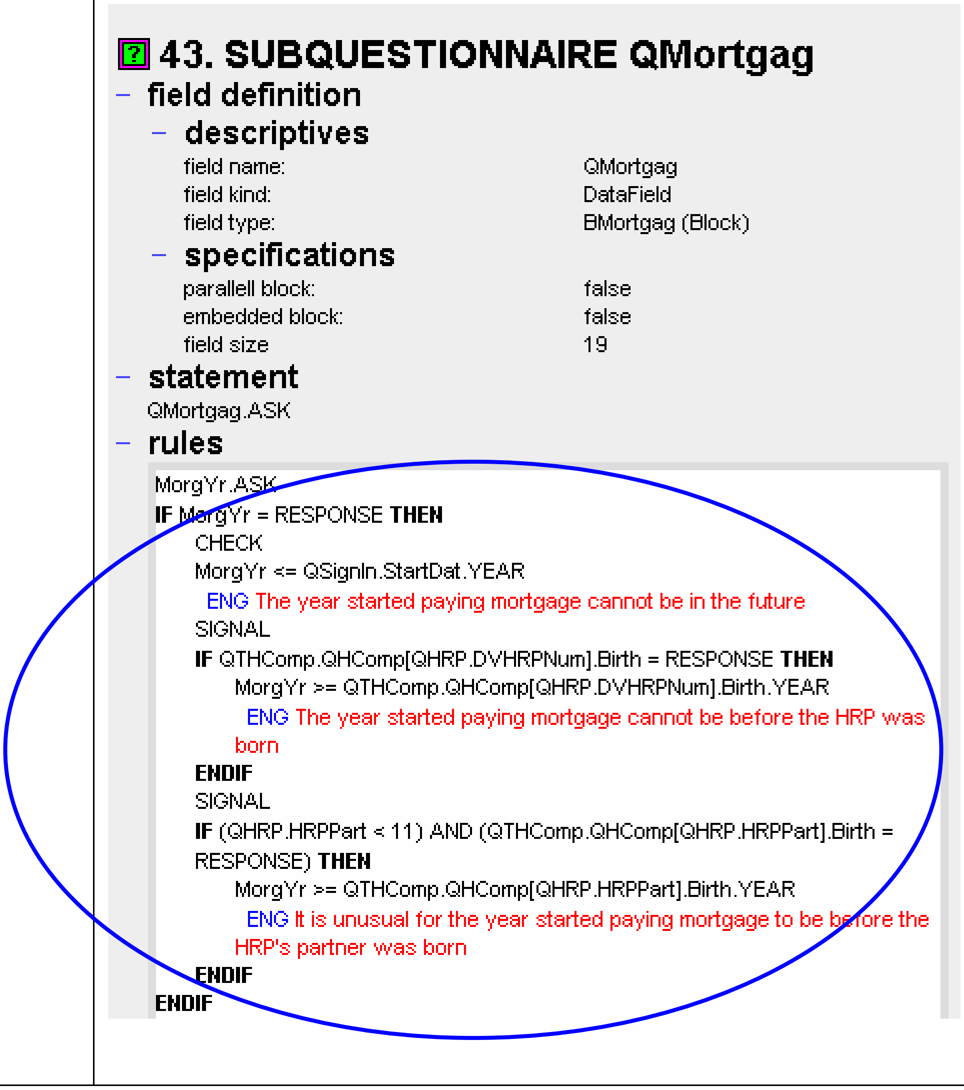
```
RULES { BNation }

 Nation
 IF (Other IN Nation)"Nation=Other" THEN
    NatSpec
 ENDIF
```



```
– rules
    Nation.ASK
    IF Other IN Nation THEN
        NatSpec.ASK
    ENDIF
```

Variable list	No option just to list all the questionnaire variables on screen. It is possible to change the default setting to display just the questions in the tree view. For example:  root. MOD0304A 1. QID 1.1. 1.1.1. IF BIDData.SEARCH (1) THEN 1.1.1.5. Area 1.1.1.6. Address 1.1.1.7. HHold 1.1.2. ELSEIF BIDData.SEARCH (2) THEN 1.1.2.1. Area 1.1.2.2. Address 1.1.2.3. HHold 1.1.3. ELSE 1.1.3.1. Area 1.1.3.2. Address 1.1.3.3. HHold	✗
Allow other information to be imported into the tool	It is not possible to import additional information into the tool.	✗

Edit checks	Checks and signals are displayed along with routing at the block level. They are not identified as a separate item in the tree view so it is not possible to specify whether they should be displayed. 	✓ ✗

43. SUBQUESTIONNAIRE QMortgag

– **field definition**

– **descriptives**

field name:	QMortgag
field kind:	DataField
field type:	BMortgag (Block)

– **specifications**

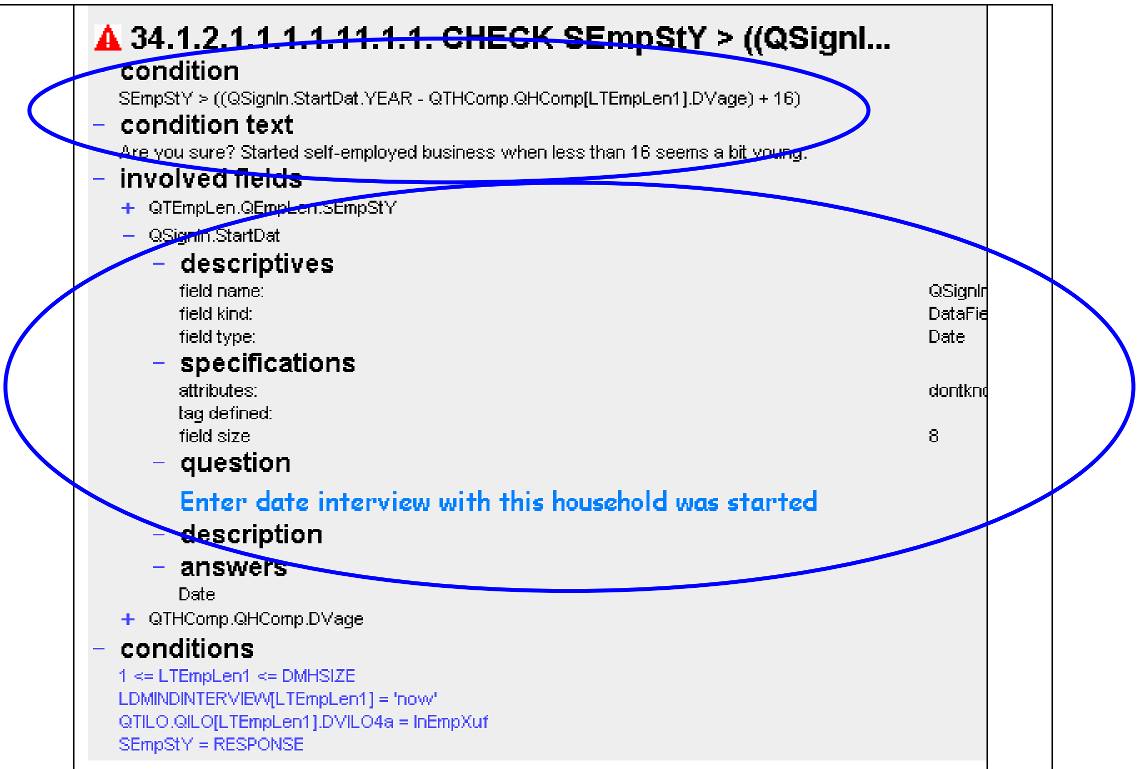
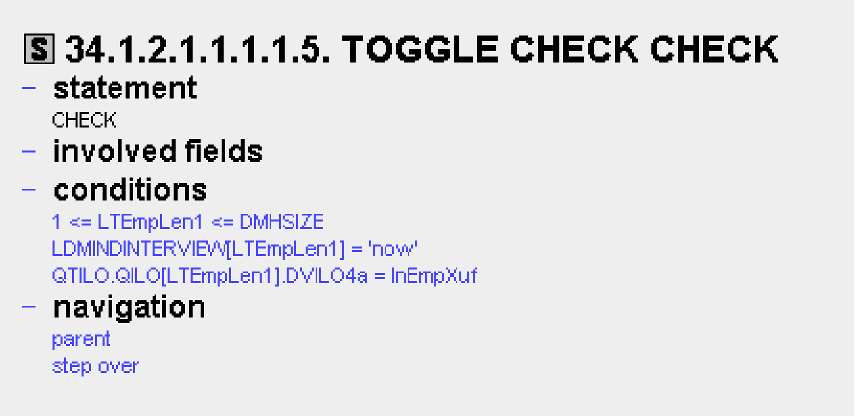
parallell block:	false
embedded block:	false
field size	19

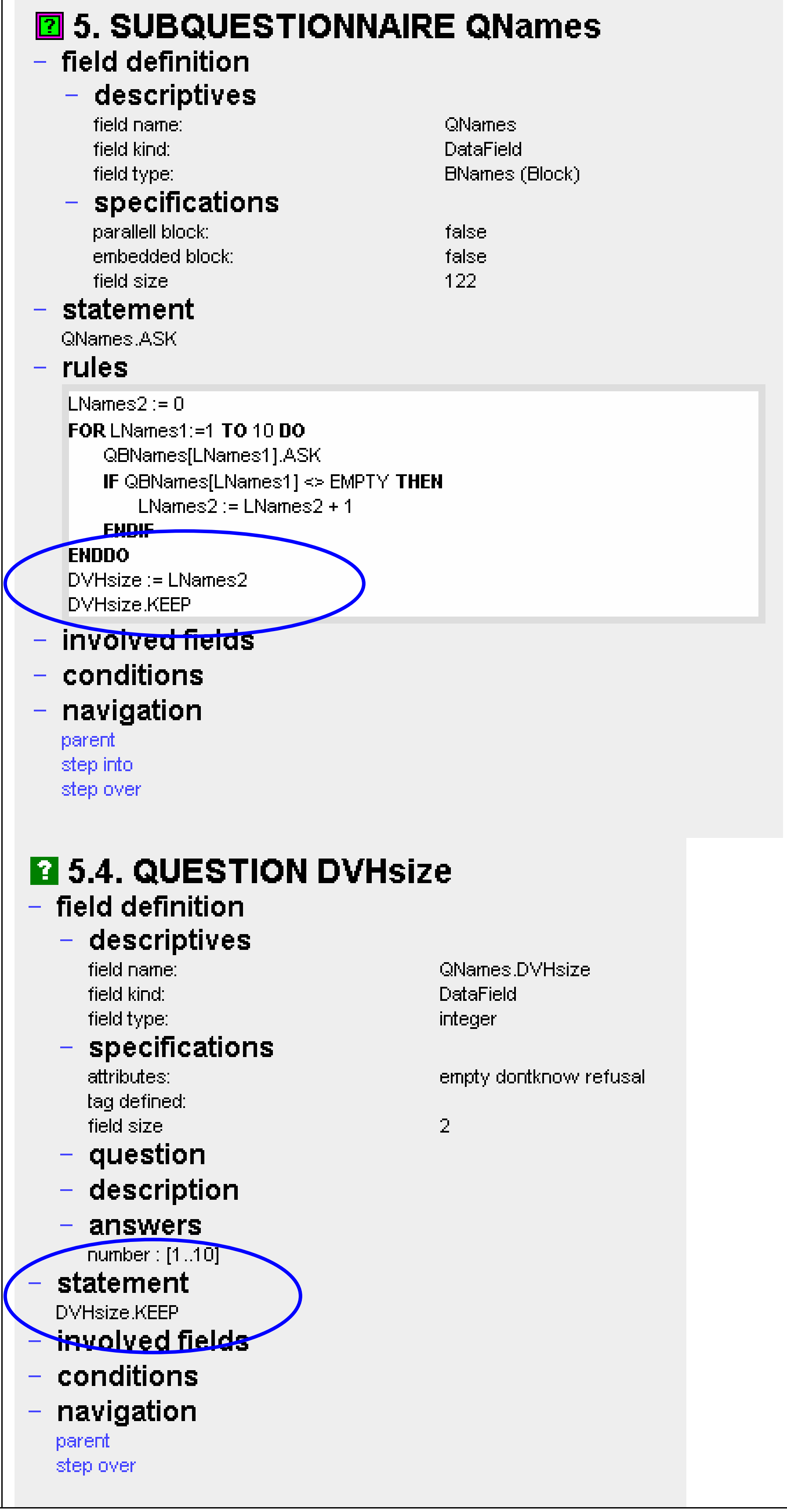
– **statement**

QMortgag.ASK

– **rules**

```
MorgYr.ASK
IF MorgYr = RESPONSE THEN
   CHECK
   MorgYr <= QSignIn.StartDat.YEAR
    ENG The year started paying mortgage cannot be in the future
   SIGNAL
   IF QTHComp.QHComp[QHRP.DVHRPNum].Birth = RESPONSE THEN
      MorgYr >= QTHComp.QHComp[QHRP.DVHRPNum].Birth.YEAR
       ENG The year started paying mortgage cannot be before the HRP was
      born
   ENDIF
   SIGNAL
   IF (QHRP.HRPPart < 11) AND (QTHComp.QHComp[QHRP.HRPPart].Birth =
   RESPONSE) THEN
      MorgYr >= QTHComp.QHComp[QHRP.HRPPart].Birth.YEAR
       ENG It is unusual for the year started paying mortgage to be before the
      HRP's partner was born
   ENDIF
ENDIF
```

	 34.1.2.1.1.1.11.1.1. CHECK SEmpStY > ((QSignI... − condition SEmpStY > ((QSignIn.StartDat.YEAR - QTHComp.QHComp[LTEmpLen1].DVage) + 16) − condition text Are you sure? Started self-employed business when less than 16 seems a bit young. − involved fields + QTEmpLen.QEmpLen.SEmpStY − QSignIn.StartDat − descriptives field name: QSignI field kind: DataFie field type: Date − specifications attributes: dontkn tag defined: field size 8 − question Enter date interview with this household was started − description − answers Date + QTHComp.QHComp.DVage − conditions 1 <= LTEmpLen1 <= DMHSIZE LDMINDINTERVIEW[LTEmpLen1] = 'now' QTILO.QILO[LTEmpLen1].DVILO4a = InEmpXuf SEmpStY = RESPONSE  S 34.1.2.1.1.1.1.5. TOGGLE CHECK CHECK − statement CHECK − involved fields − conditions 1 <= LTEmpLen1 <= DMHSIZE LDMINDINTERVIEW[LTEmpLen1] = 'now' QTILO.QILO[LTEmpLen1].DVILO4a = InEmpXuf − navigation parent step over	
Computations	Like signals and checks, computations are displayed on screen in the rules section at block level. Details about the computed variable are displayed separately. It is not possible to choose whether to display the computations.	✓ ✗



? 5. SUBQUESTIONNAIRE QNames

– field definition

– descriptives

field name:	QNames
field kind:	DataField
field type:	BNames (Block)

– specifications

parallell block:	false
embedded block:	false
field size	122

– statement

QNames.ASK

– rules

```
LNames2 := 0
FOR LNames1:=1 TO 10 DO
    QBNames[LNames1].ASK
    IF QBNames[LNames1] <> EMPTY THEN
        LNames2 := LNames2 + 1
    ENDIF
ENDDO
DVHsize := LNames2
DVHsize.KEEP
```

– involved fields

– conditions

– navigation

parent
step into
step over

? 5.4. QUESTION DVHsize

– field definition

– descriptives

field name:	QNames.DVHsize
field kind:	DataField
field type:	integer

– specifications

attributes:	empty dontknow refusal
tag defined:	
field size	2

– question

– description

– answers

number : [1..10]

– statement

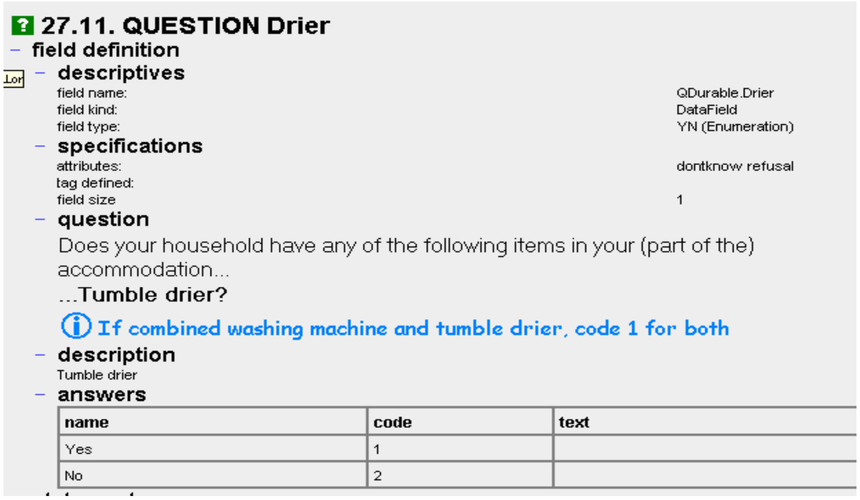
DVHsize.KEEP

– involved fields

– conditions

– navigation

parent
step over

<table>
<tr><td></td><td></td><td></td></tr>
<tr>
<td><b>How questions appear on screen</b></td>
<td>
<p>Delta does display the format of the question text, instructions and answer categories as they appear on screen. So for example if question instructions are displayed on screen in different colour to the question text this can be seen on screen.</p>
<p>However, screen colours and layout are not displayed on screen.</p>

<p><b>27.11. QUESTION Drier</b></p>
<p><b>field definition</b></p>
<p><b>descriptives</b></p>
<p>field name: QDurable.Drier
field kind: DataField
field type: YN (Enumeration)</p>
<p><b>specifications</b></p>
<p>attributes: dontknow refusal
tag defined:
field size 1</p>
<p><b>question</b></p>
<p>Does your household have any of the following items in your (part of the) accommodation...</p>
<p><b>...Tumble drier?</b></p>
<p><b>If combined washing machine and tumble drier, code 1 for both</b></p>
<p><b>description</b></p>
<p>Tumble drier</p>
<p><b>answers</b></p>
<table>
<tr><th>name</th><th>code</th><th>text</th></tr>
<tr><td>Yes</td><td>1</td><td></td></tr>
<tr><td>No</td><td>2</td><td></td></tr>
</table>
</td>
<td>✓
✗</td>
</tr>
</table>