Introduction to

# DISE

**D**YNAMIC **I**NTELLIGENT **S**URVEY **E**NGINE

Created by

Prof. Dr. Christian Schlereth

# Welcome to DISE!

DISE is an extendable, web-based survey platform. It supports technically sophisticated surveys and facilitates the development new methods for preference elicitation.

If you use DISE for scientific purposes, please include the following citation in your paper:

*Schlereth, Christian / Skiera, Bernd (2012), "DISE: Dynamic Intelligent Survey Engine", in: Quantitative Marketing and Marketing Management. Marketing Models and Methods in Theory and Practice, eds: A. Diamantopoulos, W. Fritz and L. Hildebrandt, Wiesbaden: Gabler, 225-243.*

# 1 Getting Started

Before you start with your first survey you need to go through the following list of initial set-up steps:

1. Register at www.dise-online.net
2. Log in to you newly created account
3. Download the demonstration survey, which will be a valuable resource to create all kinds of questions (cf. Figure 1). You can also access the demo version: http://www.dise-online.net/demo.aspx.
4. Download the XML scheme file dise.xsd (cf. Figure 2).
5. Save dise.xsd to the same folder as the demonstration survey and your future survey file (cf. Figure 3)
6. You may also consider downloading EmptySurvey.xml containing a simple survey consisting of just two empty pages. This file may serve as starter for your survey.

*Figure 1: Download XML demo file*

*Figure 2: Download XML scheme*
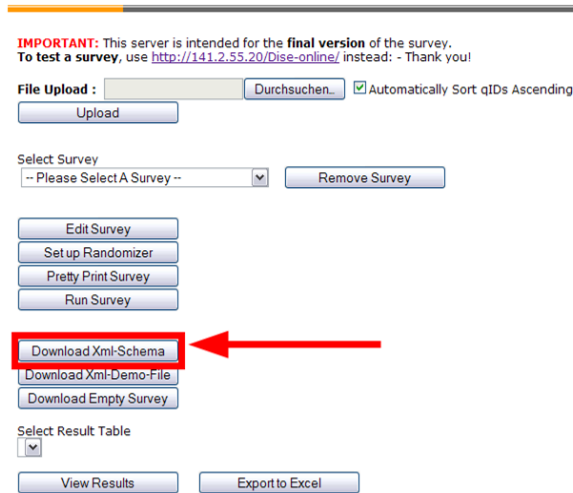


*Figure 3: Save XML demo file and XML scheme to the same folder*
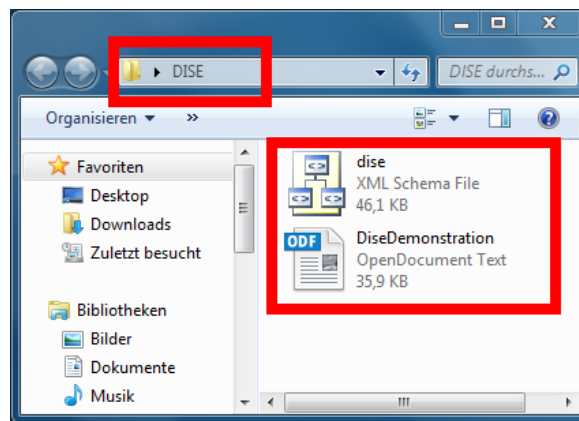


**Important stuff up front:**

- You must not at any time edit dise.xsd. It is a background programming file which may only be edited by the tool administrator. Its sole purpose is to tell your survey document, which types of questions are possible and how questions are specified. It is however recommended to have dise.xsd opened in your XML editor while crating your survey (cf. section 3.1).

- The demonstration survey is designed introduce you to all kind of possible DISE question types. If you see something in the demonstration survey that you would like to reuse, just copy and paste the respective element from the demonstration XML into your own survey document and edit the question wording and values.

- Having completed the final version of your survey upload it to DISE and open it in a browser. In the browser, the link to your survey will look more or less like that: http://www.dise-online.net/o/Survey.aspx?id=4323&pos=0&user=001104-f8403ff8-af2e-4d75-befb-f4804ff9a192

- **DO NOT DISTRIBUTE THIS LINK DIRECTLY!** Remove any information after "id=XXXX", such that it looks like this: http://www.dise-online.net/o/Survey.aspx?id=4323 and then distribute this shortened link. If you distribute the full link, every new respondent will just overwrite previous entries and you will end up with only one respondent's data.

## 2   Test Server (Sandbox)

To upload and test draft surveys, please use our test server which have access to via the links below:

- http://www.dise.sandbox.whu.edu/dise-online (off campus of WHU)
- http://192.168.160.5 /dise-online (on campus of WHU; same test server)

Only use the productive server (www.dise-online.net) after you have finished testing the questionnaire and would like to distribute it. Login information is valid for both systems such that you only need to register once.

## 3   Basics in XML

In DISE no wysiwyg-editor (for the non-nerds: wysiwyg = what-you-see-is-what-you-get) exists. Instead, surveys are directly specified as XML. XML is a markup language for representing hierarchically structured data in the form of text data.

At first sight, this might sound a little bit complicated, but it isn't! The great advantages of DISE compared to other survey platforms will be noticeable when it comes to refining your survey – typically a task that takes more time than the initial set-up. Since because there is no need of clicking through a thousands of menus DISE allows you to do this much faster than other tools. Also replicating elements from previous studies is much simpler than it would be with other tools. And an XML-editor supports you in the use of XML.

The only requirement is that you must be familiar with at least some XML basics. This chapter provides a brief overview and explanation of what you need to know and where you can get further information.

### 3.1   XML Editor Download

There are several XML-editors out there and most of them a relatively handy. So if you already have some XML experience you comfortably stick to your XML editor of choice. For all others, we'd recommend using one of the following editors:

- XML-spy or
- Liquid XML or
- EditX.

### 3.1.1   XML Spy

XML Spy is a commercial software with a free 30-days trial version.
Later screenshots are mostly taken from XML-spy. Nevertheless, other editors have similar functionalities.

You can download and obtain a key code for the trial version via: http://www.altova.com/download/xmlspy/xml_editor_enterprise.html.

### 3.1.2   Liquid XML

Liquid XML is commercial software with a free 30-days trial version.
You can download and obtain a key code for the trial version via:
https://www.liquid-technologies.com/xml-studio.

### 3.1.3   EditX XML Editor Free

For non-commercial users, EditX is a free XML editor. It is available for Windows, Linus and MacOS.
You can obtain it via:
 http://www.download3000.com/download-editix-lite--free-xml-editor-count-reg-50066.html

## 3.2   XML Basics and XML-Editor

*Starting XML editor*

Start your XML editor and open a survey file (e.g. demo survey). A survey file is opened via the following path: File→ Open -> DISEDemonstration.xml. Having done that, your XML editor will produce a screen as shown in Figure 4. On the left side of your screen, you see the tree structure of your XML document and on the right side, you see the actual content of your XML document.

*Figure 4: Opened DISE demonstration survey in EditX*
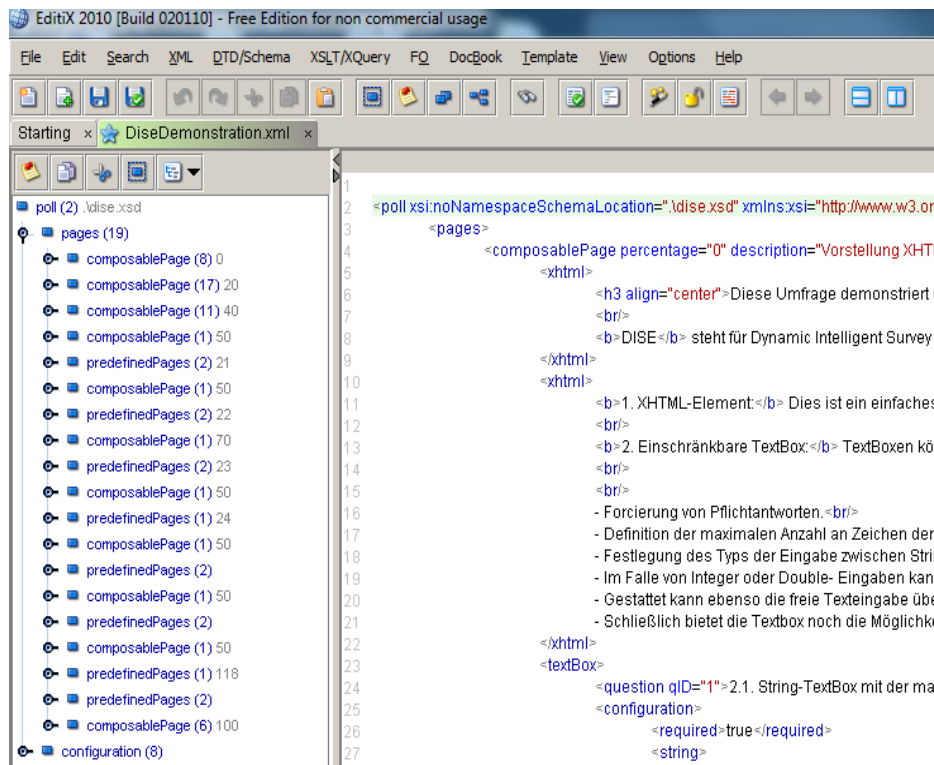


*XML-basics:*

Introduction to DISE

- XML structures information within so called tags (e.g., `<textBox></textBox>`). For example,

```
<textBox>
      <question qID="1">how old are you?</question>
</textBox>
```

- Every open tag must be closed again.
  Example:

  right: `<a>Test</a>`
  wrong: `<a>Test`

- XML structures tags in hierarchies. You can only close a tag, if all tags within this tag have been closed.

  o Example 1:

  right: `<a><b>Test<b></a>`
  wrong: `<a><b>Test</a></b>`

  o Example 2:

```
<verzeichnis>
      <titel>Städteverzeichnis</titel>
      <eintrag>
            <stichwort>Genf</stichwort>
            <eintragstext>Genf ist der Sitz
von ...</eintragstext>
      </eintrag>
</verzeichnis>
```

- Attributes are information within the tags. E.g. attribute called "id":
  `<a id="1"><b>Test<b></a>`
- DISE only knows two attributes: "`qID`" and "`markID`".
  o `qID` is used to create and link respective database columns when storing respondents' answers to particular questions. `qID` requires the specification of an integer number (>0). However, you may always set it to 1, since DISE can then automatically number your questions throughout your questionnaire.
  o `markID` is used to access information during the survey process. In contrast to `qID`, its value must be unique, before uploading the survey document to DISE. Examples in the later sections will clarify its use.

## 3.3   XHTML

For formatting purposes, it is useful to also know some basics in XHTML. XHTML is similar to HTML, the standard markup language used to create web pages. The only difference that XHTML is that it must comply with XML. This means that every opened tag must also be closed.

For example, the html-tag for break lines <br> must be written in XHTML as <br/>.

All commands of HTML are supported in XHTML. XHTML is also a great way to include images, formatted product descriptions and even videos into your questionnaire. To include an image or video, upload it to a server of your choice (e.g the WHU website server (access via Typo3)). So far, the DISE server does not support the upload of images or videos.

Most important commands to edit and your survey and give it a pretty look are the following:

- To write in bold: `<b> This text appears in bold<b>`
- To write in italics: `<b> This text appears in italics<b>`
- To underline: `<u>underline</u>`
- To draw a horizontal line: `<hr/>`
- To add an image: `<img src="http://www.linkzumbild.de/bild.jpg/"/>`
- To add a break line: `<br/>`

For more advanced formatting techniques (e.g., tables, structured lists) refer to http://www.selfhtml.org where excellent tutorials are provided.

## 3.4   Special Characters in XML-Documents

If your entry within a tag (e.g., <question>) contains special characters (cf. Table 1) remember to use the correct replacement such that DISE can correctly process the character.

*Table 1: Special characters in XHTML*

| Special character | Replacement |
|:---:|:---:|
| & | `&amp;` |
| ' | `&apos;` |
| < | `&lt;` |
| > | `&gt;` |
| " | `&quot;` |

- Example:

    <url>http://www.xhtml_tips&tricks.com</url>

    ➔

    <url> http://www.xhtml_tips&amp;tricks.com</url>

# 4   Creating a Survey

## 4.1   Survey Structure

We recommend building the survey according to a predetermined structure (cf. Figure 5).

- The beginning of the survey should include a **starting page containing name and email-address of a contact person responsible for the questionnaire.**

- The final page of the survey should be inserted at the end of the survey to thank the user for his / her cooperation. Do not include any questions in the final page!

*Figure 5: Survey structure*

Introduction to DISE
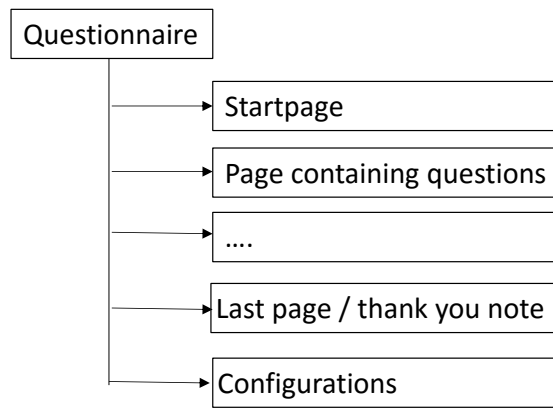


- At the end of the XML-Document, you need to include the configuration of the questionnaire (e.g. title, author, language).

XML code example: configuration of last page

```
<configuration>                                    Start of the configuration
    <title>
        #Title of the survey                       Title of the survey
    </title>

    <author>Max Muster</author>                    Citation of the author´s name and
    <email>                                        email address
        Max.Muster@Muster.de
    </email>

    <language>                                     Setting the language (German in this
        <de/>                                      case; use <en/> for English)
    </language>

    <masterPage>mymaster.master</masterPage>       Setting the Master-Page


<preventBackButton>false</preventBackButton>       Adding a "back" button on the Master-
</configuration>                                   Page
```
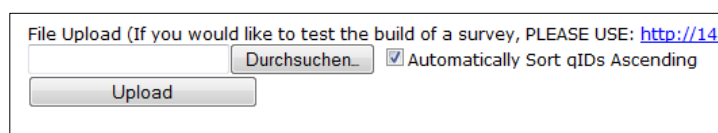
## 4.2   Uploading a Survey

To set up a survey, use the XML document, which specifies your questionnaire. Assign a distinctive name (e.g. myFirstQuestionnaire.xml) to your questionnaire and upload it to the administration area of DISE (cf. Figure 6). DISE will then create all required tables and links in the database.

*Figure 6: DISE upload function*



Having complete the survey upload, your survey will appear on the dropdown list of surveys associated with your account (cf. Figure 7).

*Figure 7: Dropdown list of surveys on your account*

Introduction to DISE



In that list, you can also find the unique survey ID (the first four numbers in each line of the list). Consistently, the link to your survey is ID based: http://www.dise-online.net/survey.aspx?id=1313

**IMPORTANT**: If you click on a survey link, DISE will automatically add a new user-ID to the link, e.g., http://www.dise-online.new/survey.aspx?id=1313&user=XXXXX. Do not distribute the link with the user-ID, or each new respondent will overwrite the answers of the previous respondent.

## 4.3   The File dise.xsd

The file dise.xsd is a scheme file that …

- … provides information about the existing questioning tags and how to configure them.
- … serves to validate and test the correctness of your questionnaire.

Save the file dise.xsd to the same folder as your questionnaire. Do not make any changes to the file; however, open it in the viewer of your XML-editor. It can be a helpful reference to all DISE functionalities.

You may for example want to search for the tag `<poll>`. To do so, you select `<poll>` on the right hand side and expand the tree (cf. Figure 8). Take your time to view all the possibilities. You will learn that a survey (`<poll>`) consists of pages (`<pages>`) which again consist of subpages (either `<predefinedPages>` see Section 5 or `<composablePages>` see Section 7). Use +/- to navigate to the desired place and see which types of questions are available (e.g. <textbox>, <conjoint>, <constant sum>) and what options for configurations they offer. Elements with solid lines indicate that the element is mandatory. Dotted elements indicate that an element is optional.

Figure 8: Tree structure oft he XML file



Click on structure Spy.  to get to the tree included only in XML
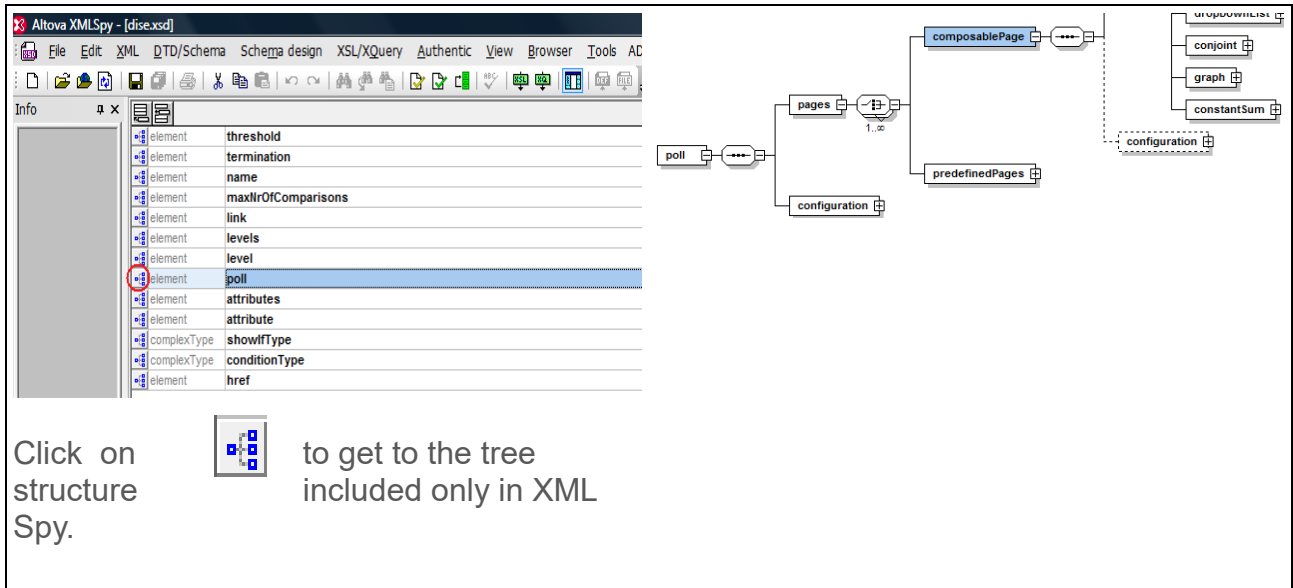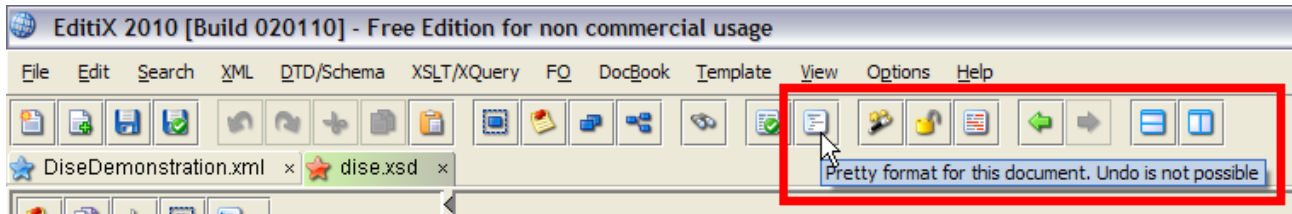
## 4.4 Creation of a new questionnaire

Open the emptyXML.xml or the demonstration xml-file in your XML-Editor. Make sure that dise.xsd has been saved to the same folder.

If you see a big block of text, use the button pretty print (cf. Figure 9) to see everything in a more structured way.

Figure 9: „Pretty print" option on EditX toolbar



## 4.5 Composable Pages vs. Predefined Pages

DISE differentiates between two types of pages:
- *Composable Pages* consist of different types of questions compiled on one single page. For example, the first page of the demonstration survey consists of a XHTML field followed by a radio-button task. The available survey items for Composable Pages are demonstrated in Figure 10.
- *Predefined Pages* are mostly advanced data collection methods, i.e., a set of predefined questions that are shown across multiple pages such as
  - o Choice-Based Conjoint (also known as traditional discrete choice experiments,
  - o Best Worst Case 1-3,
  - o SADR
  - o and many other.

  These data collection methods extend over one or more complete survey pages and follow a specific predefined process. The available data collection methods for

Predefined Pages are demonstrated in Figure 11.

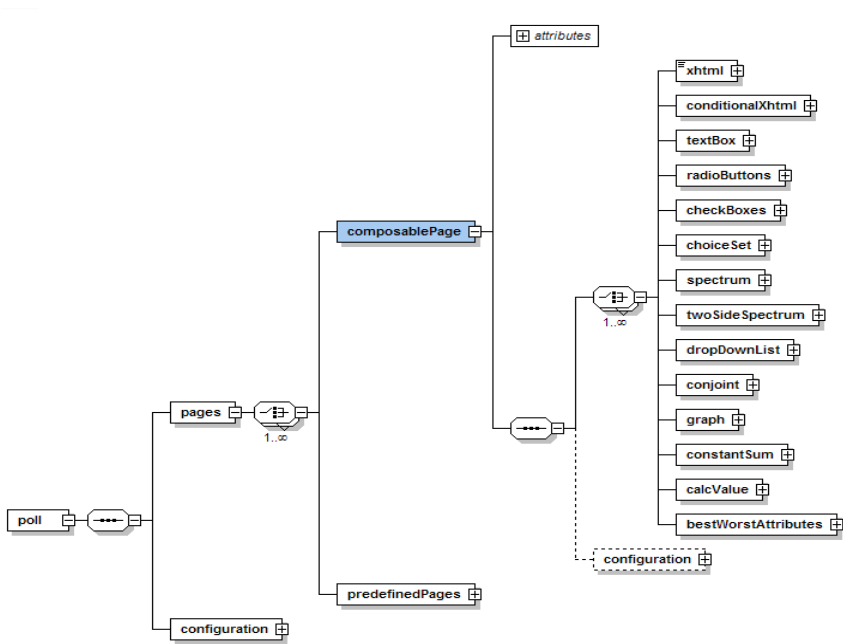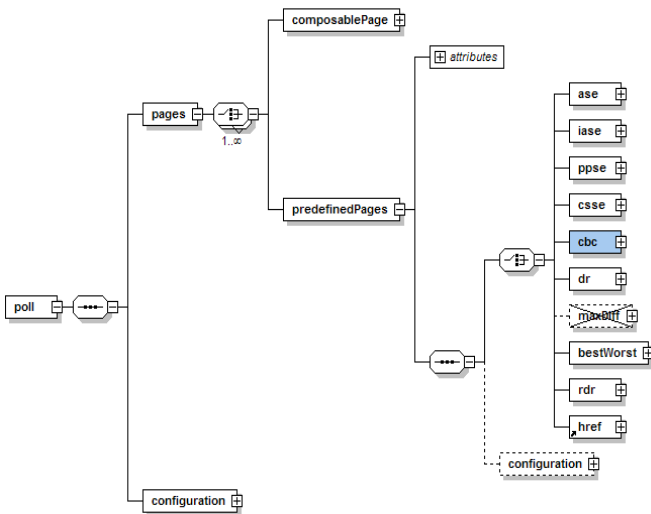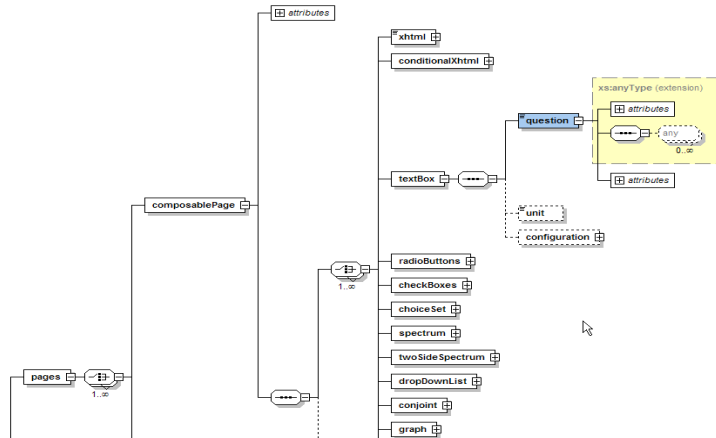*Figure 10: Available survey items for Composable Pages*



*Figure 11: Available data collection methods for Composable Pages*



## 4.6   Adding a Textbox to a Questionnaire

For all possible configurations of "textbox" refer to dise.xsd (cf. Figure 12)

*Figure 12: „Textbox" configuration in XML scheme*



The `<question>` tag must contain at least one attribute (i.e., "`qID=`") with some XHTML content (cf. section 3.3).

<u>XML code example: textbox</u>

```
<composablePage>
...
<textBox>

    <question qID="1">
        How old are you?
    </question>
        <unit>Years</unit>
        <configuration>
    <required>true</required>
            <integer>
          <minValue>0</minValue>

<maxValue>150</maxValue>
            </integer>
        </configuration>
</textBox>
...
</composablePage>
```

*Open the textbox tag.*

*Include the question. Also attach an attribute qID – there is no need to count the number of questions – always enter 1 → DISE will handle the counting.*

*Unit: Years*
*Start of the configuration*
*Should DISE force the respondent to enter an answer?*

*Specify data type as well as minimum and maximum value*

*End of the configuration*
*Close textbox tag*

The `<question>` tag can have up to two attributes: the mandatory *qID* and the optional *markID*, which serves to link the appearance of subsequent questions to previous responses (cf. section 6.1.2).

**Important:**
- `qID` requires the specification of an integer number (>0). However, you may always set it to 1, since DISE can then automatically number your questions throughout your questionnaire.
- As shown in dise.xsd, the `<question>` tag must occur directly after the opening of the `<textBox>` tag. Dashed tags (e.g. `<unit>`) are optional.

## 4.7 Forced or Optional Answers

To force respondents to provide an answer to the question, add the following line to the configurations tag:

```
<required>true</required>
```

- True: question must be answered
- False: question is optional

If you make a field a required question, respondents can only move on to the next page after filling all required fields. Otherwise, DISE will show an error message (cf. Figure 13).

*Figure 13: Error message for unfilled required question*
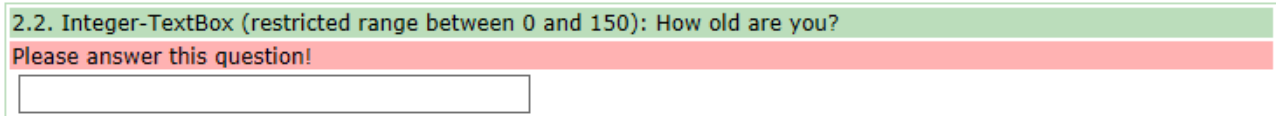


## 4.8 Percentage Progress of the Survey



The top of a survey page always includes a progress bar. Progress values have to be entered manually as an integer value between 0 and 100 in a mandatory attribute of the `<composablePage>` tag. In case of the `<predefinedPage>` tag, values are entered as ranges.

XML code example: survey progress bar

| | |
|---|---|
| `<composablePage` | *Page starts here.* |
| `percentage="0"` | *Here, the progress percentage is entered (0-100).* |
| `description="write something about the composable page...">` | *description="this is the description">* *Description of the page* |
| `</composablePage>` | *End of the page* |

## 4.9 XML check tools to ensure your questionnaire is correct

XML editors such as XML-Spy or EditX also allow for testing questionnaire with respect to format, programming errors and conformity to the assigned .xsd-file (cf. section 3).

To check for consistency and correctness, click on the icons "Check well-formedness" and „validate file".
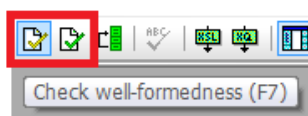
## 4.10 Clarity of the Programming Code

Another helpful icon is the "pretty print" icon, which structures the XML-content in a tree.
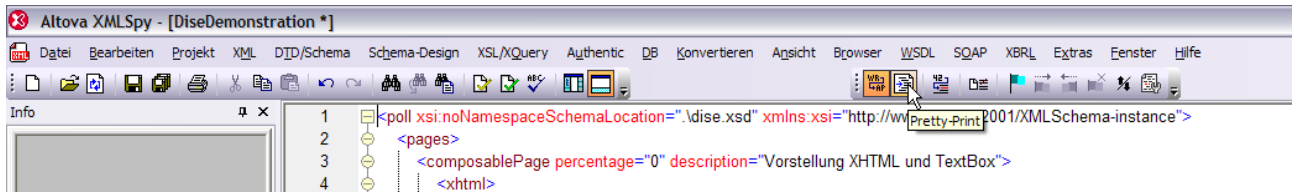


# 5 Questions types in DISE

After the introduction of textboxes in section 4.6, this section summarizes the most frequently used question types of a composablePage. Again, we refer to dise.xsd to see the all options of question types.

## 5.1 XHTML

Content (e.g., instruction texts, images, videos, …) are integrated within the XHTML tag:

- Example:

```
<xhtml>
     This is some text, which contains <b>bold words</b> and
     <u>underlined words</u>. If you would like to enter a line
     break, just include: <br/> If you want to include images use
          <img src="http://www.dise-
     online.net/images/Logo_MES.png"/>
</xhtml>
```

## 5.2 Text Boxes

2.1. String-TextBox with a maximum of 20 characters: What is your name?

TextBoxes can be configured quite flexibly. For example, you can insert text directly after a textBox using `<unit> (e.g., €, $, …)</unit>`.

If you want to ensure the correct format of answers, include one of the following tags into `<configuration>`:

### 5.2.1 Integer

| | |
|---|---|
| `<integer>` | *Specification: finite range of integer values* |
| `<minValue>0</minValue>` | *Value of minimum  0…* |
| `<maxValue>150</maxValue>` | *…and maximum 150* |
| `</integer>` | *End.* |

## 5.2.2 Decimal

```
<decimal>

    <minValue>0</minValue>

    <maxValue>150</maxValue>

</decimal>
```

*Specification: Decimal numbers*

*Value of minimum 0…*

*…and maximum 150*

*End.*

Please check out dise.xsd or the textboxes in the demonstration file to understand all the different possibilities. For example, it is possible to specify the width of a textbox, the number of lines it supports, and much more.

## 5.3 Radio Buttons

RadioButtons is a question type suitable for forcing the respondent to decide on one alternative among a predefined set of possible alternatives. Buttons are arranged either vertically or horizontally.



XML code example: radio buttons

```
<radioButtons>

  <question qID="1">Please decide
between:</question>

  <items>

    <item>Option 1</item>

    <item> Option 2</item>

    <item> Option 3</item>

  </items>

<configuration>

  <required>true</required>

  <renderVertically>true
</renderVertically>

</configuration>

</radioButtons>
```

*Question type (here: radioButtons)*

*Type the question here.*

*Start of the selection*

*Option 1*

*Option 2*

*Option 3*

*End of the selection*

*Start of the configuration*

*Make question required or not (here: required)*
*Choose optical arrangement (here: vertical)*

*End of the configuration*

*End*

## 5.4   Check Boxes

CheckBoxes are similar to radio buttons, but enable respondents to select multiple alternatives.

| 4.1 Horizontale CheckBoxes: Was trifft zu? | | |
|---|---|---|
| Möglichkeit 1 ☐ | Möglichkeit 2 ☐ | Möglichkeit 3 ☐ |

**4.2 Vertikale CheckBoxes: Was trifft zu?**
☐ Möglichkeit 1
☐ Möglichkeit 2
☐ Möglichkeit 3

The specification of CheckBoxes is similar to RadioButtons. The only difference is that the attribute `qID` is within the `<item>` tag and not within the `<question>` tag to store the multiple decisions.

## 5.5   Choice Sets

ChoiceSets are a popular way to collect choice decisions and facilitate reserchers' understanding of respondents' decision making.

The presented example (cf. Figure 14)  shows three product alternatives with different attributes and levels. If none of the alternatives provides sufficient utility, the respondent can also pick the no-choice option.

*Figure 14: ChoiceSet*

| Which of the following MP3-Players do you prefer most?: | | | | |
|---|---|---|---|---|
| Brand | Apple | Samsung | Sony | I would buy |
| Storage capacity | 20 GB | 4 GB | 1 GB | none |
| Price | 299 ? | 100 ? | 120 ? | of these three |
| | | | | |
| | ○ | ○ | ○ | ○ |

Choice Sets can be entered manually using the `<choiceSet>` tag. The `<predefinedPage>` also contains a choice based conjoint element (also known as discrete choice experiment) `<cbc>` which facilitates the specification and execution of these types of experiments.

Introduction to DISE

<u>XML code example: choice set</u>

```
<choiceSet>
<question qID="12">Which of the following MP3-
Players do you prefer most?:</question>

<definition>
    <attributeDef>Brand</attributeDef>
    <attributeDef>Storage
Capacity</attributeDef>
    <attributeDef>Price</attributeDef>
    <attributeDef> </attributeDef>

</definition>

<items>
    <item>
        <levelValue>Apple</levelValue>
        <levelValue>20 GB</levelValue>
        <levelValue>299 €</levelValue>

        <levelValue><center>
        <img alt=""
src="../userupload/ex_ipod.jpg"
height="40px"/>
        </center></levelValue>

    </item>

    <item>
    …
    </item>

    <item>
        <levelValue>I would buy</levelValue>
        <levelValue>none </levelValue>
        <levelValue> of these three
</levelValue>
    </item>
</items>
</choiceSet>
```

*Question type (here: ChoiceSet)*

*Start of attribute definition*

*End of attribute definition*

*Start of items*

*First item (here: attribute1 level 1)*

*Centering the included image*

*Image source*

*End of centering. End of level value*

*End of first item*

*Start second item (cf. first item)*

*End of the second item*

*Start third item (no-choice option)*

*End of no-choice-option*
*End of items*
*End of ChoiceSet*

## 5.6   Ranking Based Analysis

Another possibility is to rank products described by their attributes and levels using the conjoint element. The maximum number of the conjoint items is restricted to 30 (cf. Figure 15).

*Figure 15: Conjoint option in DISE*

<u>XML code example: conjoint</u>

```
<conjoint>
<question>Sort the following elements in a
ranking:</question>

<definition>
    <attributeDef>Brand</attributeDef>
    <attributeDef>Storage
Capacity</attributeDef>
    <attributeDef>Price</attributeDef>
    <attributeDef/>
</definition>

<items>
    <item>
        <levelValue>Apple</levelValue>
        <levelValue>20 GB</levelValue>
        <levelValue>299 €</levelValue>

        <levelValue><right>
            <img alt=""
            src="../userupload/ex_ipod.jpg"
            height="40px"/>
        </right></levelValue>

    </item>

    <item>
        ...
    </item>

</items>

<configuration>
    <required>true</required>
    <nrOfCols>3</nrOfCols>
</configuration>

</conjoint>
```

| Code | Description |
|------|-------------|
| `<conjoint>` | *Question type (here: conjoint)* |
| `<question>` | *Enter the question here.* |
| `<definition>` | *Start of the domain* |
| `<attributeDef>` | *Different attributes on the left side of the display tables* |
| `</definition>` | *End of domain* |
| `<items>` | *Start of items* |
| `<item>` | *Start of first Item* |
| `<levelValue>` | *Enter the different attribute levels here.* |
| `<right>` | *Set image left-justified* |
| `<img>` | *Image source* |
| `</right>` | *End of setting image left-justified* |
| `</item>` | *End of first item* |
| `<item>` | *Start of second item (cf. first item)* |
| `</item>` | *End of second Item* |
| `</items>` | *End of Items* |
| `<configuration>` | *Start of configuration domain* |
| `<required>` | *Make question required* |
| `<nrOfCols>` | *Number of the stimuli per row* |
| `</configuration>` | *End of configuration domain* |
| `</conjoint>` | *End of Conjoint* |

## 5.7  Spectrum

The spectrum element may be used implement standard likert scales. Such scales are especially common experimental researchers in marketing, psychology and sociology whenever people's attitudes and opinions are the research interest (cf. Figure 16).

*Figure 16: Likert scale designed using the DISE Spectrum function*

| What do you think about the following statements? | Agree | Slightly agree | Neither agree nor disagree | Slightly disagree | Disagree |
|---|---|---|---|---|---|
| Statement 1 | ○ | ○ | ○ | ○ | ○ |
| Statement 2 | ○ | ○ | ○ | ○ | ○ |

Introduction to DISE

XML code example: spectrum

```
<spectrum>
<question>What do you think about the
following statements</question>

<scale>
    <item>Agree</item>
    <item>Slightly agree</item>
    <item>Neither agree nor disagree</item>
    <item>Slightly disagree</item>
    <item>Disagree</item>
</scale>

<items>
    <item qID="1">Statement 1</item>
    <item qID="2">Statement 2</item>
 </items>

<configuration>
...
</configuration>
</spectrum>
```

*Question type (here: Spectrum)*
*Enter instructions / question text here.*

*Start of scale determination*
*Each item represents a nuance of agreement.*

*End of scale determination*

*Start of questions*
*Start of first question*

*End of the questions*

*Configuration tag where you make the question required etc.*

*End Spectrum*

## 5.8 Two Side Spectrum

Another option similar to Spectrum is the TwoSideSpectrum:

| Please evaluate the previous tasks: | | | | | | |
|---|---|---|---|---|---|---|
| Simple | ○ | ○ | ○ | ○ | ○ | Complex |
| Pleasant | ○ | ○ | ○ | ○ | ○ | Unpleasant |
| Easy | ○ | ○ | ○ | ○ | ○ | Hard |
| Confusing | ○ | ○ | ○ | ○ | ○ | Clear |
| Easy to follow | ○ | ○ | ○ | ○ | ○ | Hard to follow |
| Difficult to complete | ○ | ○ | ○ | ○ | ○ | Not difficult to complete |

In contrast to Spectrum, the respondent must calibrate his / her opinion beween two extremes.

You may also vary the number of radio buttons in the configuration of TwoSideSpectrum. This is done in the configuration area using the `<nrRadioButtons>` tag.

XML code example: two side spectrum

```
<nrRadioButtons>5</nrRadioButtons>
```

*Here, a five step scale is chosen.*

## 5.9   Best Worst

The Best Worst Task (Case 1 and 2) allows for the evaluation of the most and least preferred attribute (cf. Figure 17).

*Figure 17: Best-Worst Task in DISE*

| Bla Bla Bla | | |
|---|---|---|
| Most important | Property | Least important |
| ○ | asdf | ○ |
| ○ | fdsa | ○ |
| ○ | fdas | ○ |

## 5.10  Drop Down List

Drop down lists are similar to radio buttons. They only allow the respondent to choose one answer from a predefined list of answers (cf. Figure 18).

*Figure 18: Drop down list in DISE*

```
--- Please make your selection ---
Possibility 1
Possibility 2
Possibility 3
```
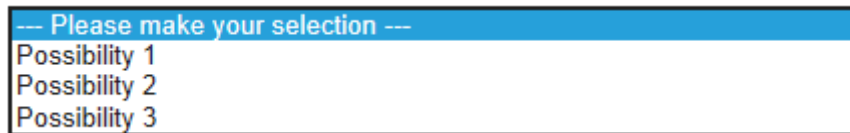
XML code example: dropDownList

| | |
|---|---|
| `<dropDownList>`<br>`<question qID="1"/>` | *Question type (here: DropDownList)* |
| `<items>` | *Start of item selection* |
| `    <item>--- Please make your selection ---`<br>`</item>`<br>`    <item>Possibility 1</item>`<br>`    <item>Possibility 2</item>`<br>`    <item>Possibility 3</item>`<br>`</items>` | *Start of items*<br><br><br><br><br>*End of items* |
| `<configuration>`<br>`    ...`<br>`</configuration>` | *Configuration tag where you make the question required etc.* |
| `</ dropDownList >` | *End of dropDownList.* |

# 6   Control the Display of Questions and Contents

## 6.1   Show if (on pages and Survey-Artifact-Level)

The ShowIf command provides the possibility to show a question or content depending on previous answers. It can be combined with almost all survey elements. For example, it only makes sense to ask a respondent about his driver's license if the respondent stated in a previous question that he / she was over 16.

To apply the showIf command, the question you refer to and the question containing the reference must appear on separate pages, such that the later question (here: about the driver license) can access the information about age.

Introduction to DISE

## 6.1.1 Insertion of a MarkID

Insert a MarkID next to the qID into all question you would like to refer to later on (e.g., the age question).

XML code example: mark ID insertion

```
<textBox>
<question qID="1" markID="1">2.2. Integer-TextBox
(restricted between 0 and 150): How old are
you?</question>

<configuration>
    <required>true</required>
    <integer>
        <minValue>0</minValue>
        <maxValue>150</maxValue>
    </integer>
</configuration>

</textBox>
```

*Question ID + MarkID*

The number used within the MarkID must be unique and set manually. When uploading the xml-document, DISE will check all mark IDs' uniqueness and will return an error message if mark IDs are not unique.

## 6.1.2 Referring to the MarkID

Having inserted the mark ID, it is now possible to refer to respondents' answers provided in the age question in later questions. For example, if the question about the driver license should only be shown to respondents older than 18, you just have to add the showIf tag in the `<configuration>` tag and refer to it using the mark ID.

XML code example: mark ID reference

```
<textBox>
<question qID="30">Ok, you stated
tob e 18 years or older. For how
many years do you own a driver
license?</question>

<configuration>
    <showIf>
        <condition>
            <markID>1</markID>
            <greaterEqualThan/>
            <value>18</value>
        </condition>
    </showIf>
</configuration>

</textBox>
```

*Start of the showIf condition:*

*You are referring to the question containing mark ID 1. Respondents see this question if and only if the provided a value greater than 18 in the referred question.*

**Mark ID allows for the following comparisons:**

| | | |
|---|---|---|
| `<equal>` | = | Is equal to |
| `<lessThan>` | < | Is less than |
| `<lessEqualThan>` | <= | Is less or equal to |
| `<greaterThan>` | > | Is greater than |
| `<greaterEqualThan>` | >0 | Is greater or equal to |
| `<notEqual>` | ≠ | Is not equal to |
| `<quotaExceeded>` | | this is currently only for testing purpose |

You can extend the conditions, by adding an <and> or an <or>-Tag. DISE tests each condition sequentially until the first (set of) conditions are fulfilled.

## 6.2 Conditional Forwarding

The ConditionalForwarding command is quite similar to the ShowIf. Using ConditionalForwarding, respondents may be forwarded to a different internal survey page (also marked via markID) or to an external website.

### 6.2.1 Insert a MarkID

Insert a MarkID next to the qID into the question you would like use as a starting point for your forwarding logic (e.g., the age question).

XML code example: insertion of a mark ID

```
<radioButtons>

<question qID="1" markID="1000"> Which of the        Insert the markID here.
following advanced data collection methods would you
like to test?</question>

<items>
    <item>CBC (Choice-Based Conjoint-Analyse, also
known as traditional Discrete Choice Experiment, as
proposed by Louviere and Woodworth 1983)</item>
    <item>Mouselab (Different configuration of a
Mouselab Experiment, as proposed by Johnson, Payne,
Schkade, and Bettman 1989)</item>
      <item>Best-Worst Discrete Choice Experiment,
Case 2, as proposed by Flynn et al. (2007)</item>
      <item>Best-Worst Discrete Choice Experiment,
Case 3, as proposed by Louviere et al. (2008) and
Louviere and Islam (2008</item>
      <item>…</item>
                            <
</items>

<configuration>
    <required>true</required>
    <renderVertically>true</renderVertically>
</configuration>

</radioButtons>
```

## 6.2.2   MarkID setting on Page-Level

Now, set another markID to the page or page level you'd like to forward to.

```
<predefinedPages markID="12">
</predefinedPages>
```

or

```
<composablePage percentage="50" markID="12">
</composablePage>
```

## 6.2.3   Setting the <configuration> tag

As a last step, include the forwarding `<configuration>` tag on the destination page at the end of question / page from where your forwarding logic starts off.

XML code example: forwarding within survey

```
<conditionalForwarding>
    <condition>
        <markID>1000</markID>
        <equal/>
        <value>2</value>
    </condition>

<destinationPageMarkID>12</destinationPageMarkID>
</conditionalForwarding>
```

*Condition:*
*When markID1000*

*equal to 2, then*

*forward to page with MarkID 12*

XML code example: forwarding to external sites

```
<conditionalForwarding>
    <condition>
        <markID>1000</markID>
        <equal/>
        <value>5</value>
    </condition>

    <href>
        <url>http://www.faz.net</url>

        <configuration>
            <addPassID>true</addPassID>
        </configuration>
    </href>
</conditionalForwarding>
```

*Condition:*

*When markID1000*

*equal to 5*

*forward to URL*

## 6.3   CalcValue

Using the CalcValue tag, you can perform calculations and show respondents a previous response. To do so, assign a markID the question you wish to refer to later on.

For example, you might want to infer the volume of data that a respondent has used for internet applications during the past year. To do so, you can elicit different numbers (e.g. number of e-mail sent per day etc.) Using CalcValue you can easily compute each respondent's data use.

Introduction to DISE

## XML code example: CalcValue

```
<textBox>
<question qID="xyz" markID="1">On average, how many emails do you send
and receive per day?<question>
<unit>Number of emails</unit>

<configuration>
    <required>true</required>
    <width>50</width>
    <integer>
        <minValue>0</minValue>
    </integer>
</configuration>
</textBox>
```

Applying the following assumptions, you can now compute each respondent's annual data volume:

- One E-Mail causes an average data volume of 0,05MB.
- The respondent claims to send 26 Mails a day.

## XML code example:

| | |
|---|---|
| ```<calcValue>``` | *Start of calcValue* |
| ```<preXhtml> We've predicted that your email traffic is about </preXhtml>``` | *Computation output* |
| ```<expression>``` | *Start of computation* |
| ```    <markID>1</markID>``` | *Value of question 1 multiplied by the volume per E-mail.* |
| ```    <mult/>``` | |
| ```    <value>0.05</value>``` | |
| ```    <mult/><value>30</value>``` | *Multiplied by the number of days in a month.* |
| ```    <mult/><value>12</value>``` | *Multiplied by 12 for a year.* |
| ```  </expression>``` | *End of computation* |
| ```<postXhtml> MB per year</postXhtml>``` | *Insert "unit"for computation output* |
| ```<configuration>``` | *Output configuration bold, green font color* |
| ```    <style>color:green; font-weight:bold</style>``` | |
| ```</configuration>``` | |
| ```</calcValue>``` | *End calcValue* |

## 6.4 Randomizer

Sometimes, it is quite useful to implement different survey versions and to assign respondents randomly to one of the versions. To accomplish that, upload all versions of your survey to DISE, go to DISE administration page and open the randomizer (cf. Figure 19).

Figure 19: Randomizer selection
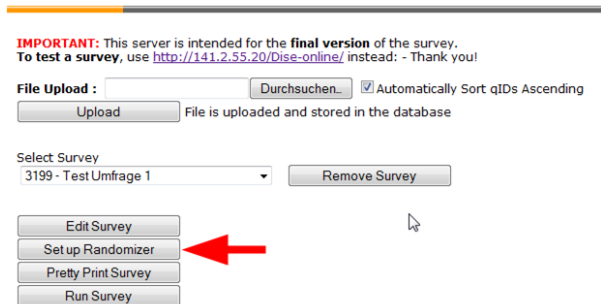


Having chosen the "Set up Randomizer" option, a pop-up window will open. You can now select all surveys you'd like to include into the randomization. To select multiple entries, hold down the Ctrl key and click on the required surveys. As a last step, you are required to enter a name for your set of randomized surveys save you selection (cf. Figure 20). When setting a name for your selection of randomized surveys, avoid blanks and special characters (e.g. right: GreatSurvey; wrong:  Great Survey#!).

DISE will show you the link to your randomized selection. You also have the possibility to store the link under your favorites. The link should display as follows: *http://dise-online.net/surveystart.aspx?config=**GreatSurvey***.

Figure 20: Selection several surveys for randomization
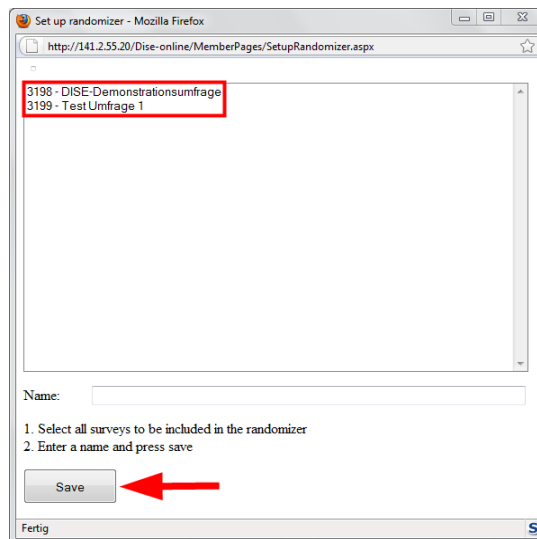


## 6.5 Integration of Online Panel Providers

Furthermore, it is possible to integrate online panel providers into DISE studies. It is even possible to integrate DISE into an Amazon turk study. The major purpose of these online panel providers is to acquire survey respondents to complete the questionnaire. Online panel providers send a link to their respondents. That link will then forward respondents to the underlying DISE survey. To control questionnaire completion, providers typically add a keyword to the DISE-link (URL) and the ID of the respondent. For example, in the following link, the keyword is "tic" and the ID is 12345678:

Introduction to DISE

Online panel providers demand that ID upon survey completion or whenever the respondent does not meet certain characteristics. The following XML code demonstrates how this is achieved.

**Important:** Concerning the pasted URLs, it should be noted that the special characters should also be replaced as described in Table 1.

To integrate Online Panels in surveys, integrate the following tags into the configuration at the bottom of your survey XML.

XML code example:

```
<configuration>…


<passIDQueryString>tic</passIDQueryString>
    <returnPassIDQueryString>
        return_tic
    </returnPassIDQueryString>…

</configuration>
```

*Online panel ID*

*Return of that specific ID*

Now, DISE will look for the keyword "tic" and return IDs upon finding the keyword return_tic.

If respondents are to be screened out (within a composablePage – for example because of little interest in the studied product) insert the following code at the end of the composablePage.

XML code example:

```
<configuration>
    <conditionalForwarding>
        <condition>
            <markID>3</markID>
            <equal/>
            <value>3</value>
        </condition>

        <href>
            <url>
                www.Link_Completed_for_PanelProvider.com
            </url>

            <configuration>
                <addPassID>true</addPassID>
            </configuration>
        </href>
    </conditionalForwarding>
</configuration>
```

*A case in which the respondent should be screened out.*

*Forward to*

*URL*

*Return or the ID*

To indicate that the respondent successfully completed the questionnaire, insert a new final predefinedPage.

Introduction to DISE

XML code example:

```
<predefinedPages>
    <href>
        <url>
            www.Link_Completed_for_PanelProvider.com
        </url>

        <configuration>
            <addPassID>true</addPassID>
        </configuration>
    </href>
</predefinedPages>
```

*Return of the ID*

**Important:** In this case, do not show a thank you page to ensure that the respondent is forwarded.

# 7   Elements of Predefined Pages

## 7.1   Choice Based Conjoint Analysis



The primary goal of DISE is to simplify the execution of advanced data collection methods such as Choice Based Conjoint. Instead of manually specifying a set of choice sets, the `<cbc>` tag defines a set of attributes, their respective levels, and the experimental design. Based on that information, DISE automatically creates the required database fields and handles the display of choice-sets on the website.

XML code example:

```
<cbc percentageStart="25" percentageEnd="90">
<introductionXhtml>This example shows a
traditional discrete choice-experiment, also
known as choice-based
conjoint.<br/></introductionXhtml>
<choiceSetQuestion>Please select the most
preferred alternative</choiceSetQuestion>
<attributes>
    <attribute>
        <name> Brand</name>
        <isNominal/>
        <levels>
            <level>
                <text>Apple</text>
            </level>
            <level>
                <text>Sony</text>
            </level>
            ...
        </levels>
```

*Start of CBC: Start & End (%)*
*Enter question text here.*

*Start of attributes*
*Attribute 1*
*Attribute name*
*Declaring attribute nominal*
*Start of attribute levels*
*Start of level 1*
*Level 1 name*
*End of level 1*
*Start of level 2*
*Level 2 name*
*End of level 2*
*etc. for several levels*
*End of attribute levels*

Introduction to DISE

```
     </attribute>
...
<cbcDesign>
...
</cbcDesign>

<configuration>

<noChoiceSetsPerPage>3</noChoiceSetsPerPage>

<noProductsPerChoiceSet>3</noProductsPerChoi
    ceSet>
    <hasNoChoice>true</hasNoChoice>
    <textNoChoice> I would not buy any of the
three products</textNoChoice>

<storeDurationPerPage>true</storeDurationPer
    Page>

<noSortedCompleteFactorialDesign>true</noSor
    tedCompleteFactorialDesign>
</configuration>

</cbc>
```

*Determination of the CBC design in detail – see below*
*End of the design*

*Start of configuration*
*Number of choice sets per page*

*Number of product per choice set*
*No-purchase-option Yes/NO*
*Text for the no-purchase-option*

*Save time respondents spend on each choice*

*End of configuration*

*End of CBC*

The CBC design can directly be inserted into DISE via copy and paste from Excel or any design generator (e.g. sawtooth, XML spy) (cf. Figure 21).

*Figure 21: Insertion of CBC design from sawtooth or XML-spy*

2,2,3;
1,1,1;
3,2,1;
3,1,2;
2,3,2;
3,1,3;
3,1,2;
2,2,1;
1,1,3;
1,3,2;
2,3,1;
3,1,3;
3,2,1;
3,3,2;
2,1,3;
3,1,1;
2,2,3;
1,1,2;

## 7.2   Dual Response

DISE is one of the few tools to support a wide range of different methods of choice based conjoint analysis. To name only some of them, you can choose between Dual Response (cf. Figure 22 and code example below), Best Worst, SADR (Separated adaptive dual response) and many other types of discrete choice experiments.

All those methods are implemented similar to the standard CBC. Differences only affect the `<configuration>` tag. For more detailed information on all methodological possibilities, check dise.xsd or the demonstration survey.

*Figure 22: Dual response with DISE*

| 1 | | | |
|---|---|---|---|
| Brand | Samsung | Apple | Sony |
| Storage capacity | 8 GB | 20 GB | 8 GB |
| Price | 299 € | 100 € | 100 € |
| Which of these products do you prefer most | ○ | ○ | ○ |

Would you buy any of the three products?

○ Yes

○ No

XML code example: Dual Response method

```
<composablePage>
    <predefinedPages>
        <dr percentageStart="55"
percentageEnd="100">

            <choiceSetQuestion>Please select the
most preferred alternative </choiceSetQuestion>

            <purchaseQuestion>Would you buy any of
the three products?</purchaseQuestion>

                <attributes>
            <attribute>
                <name>Brand</name><isNominal/>
                <levels>
                    <level>
                        <text>Apple</text>
                    </level>
                    <level>
                        <text>Samsung</text>
                    </level>
                    ’’’
                </levels>
            </attribute>

            <attribute>…
            </attribute>
        </attributes>
```

Initiation of the Dual Response method

First question

Second question

Definition of attributes
1. Attribute
Attribute type

1.     Attribute level

2.     Attribute level

2. Attribute

…

```
                    <design>
                    2,2,3;
                    1,1,1;
                    3,2,1;
                    3,1,2;
                    2,3,2;
                    3,1,3;
                    3,1,2;
                    2,2,1;
                    1,1,3;
                    1,3,2;
                    2,3,1;
                    3,1,3;
                    3,2,1;
                    3,3,2;
                    2,1,3;
                    3,1,1;
                    2,2,3;
                    1,1,2;
                    </design>
```

*Design-Matrix (see cbc)*

```
<configuration>
  <noChoiceSetsPerPage>3</noChoiceSetsPerPage>
  <noProductsPerChoiceSet>3</noProductsPerChoiceSet>
</configuration>
</dr>
```

*configuration (similar to cbc)*

## That's it.
## Enjoy DISE and good luck with your study!