

# Triple-S Version 3.0 Release Notes

## Introduction

The new features introduced into the Triple-S Standard at Version 3.0 mostly reflect developments in the nature of survey data and metadata in the industry since the last version was released, that is in the last nine years, since 2006. In some cases, exporters have in fact incorporated some of these features in their exports, and so these changes are intended to legitimate some of those practices. The new features are:

- Support for data in UTF-8 format
- Support for html formatting in text (“rich text”)
- Support for multiple keys in hierarchical data
- Declaration of a default language for multi-language surveys
- Support for scores in variables of type “multiple”
- Support for value code 0, and literal value codes, in spread multiples

## UTF-8 Data

In the original, “classic”, version of Triple-S, both data and metadata used the ISO-8859-1 standard to define the character set to be used. When XML was introduced at Triple-S version 1.1, the encoding of the metadata file was determined by the “encoding” attribute in the first line of the file, and the default encoding (if the attribute was absent) was assumed to be UTF-8. The encoding of the data remained fixed as ISO-8859-1.

With Triple-S version 3.0 we are introducing the ability for exporters to write, and importers to read, data encoded in UTF-8 format. This is specified in a new “encoding” attribute in the <record> element, viz:

```
<record ident="A" encoding="UTF-8">
```

Note that this is the only additional encoding allowed for data in version 3.0, other than the default encoding. Other encodings, e.g. UTF-16, are not allowed, but UTF-8 does cover all characters within other data encodings.

Both fixed format and CSV data may be encoded in UTF-8. Also there is no requirement that the encoding used for the XML metadata file and the data file should match.

We have also changed the default encoding for data. We found that in practice some exporters were inadvertently using the Windows-1252 encoding for data. Windows-1252 differs from ISO-8859-1 only in that ISO-8859-1 defines the code points 128-159 as control characters, whilst Windows-1252 defines them as displayable characters (including the euro symbol). As data should not include control characters, this change should have no effect on anyone exporting data in ISO-8859-1, but will now permit the legal export of data including the Windows-1252 extra characters. For importers the change means that they should now accept data that includes these extra characters.

For importers who do not support UTF-8 data it may still be possible to read the data file, retaining as many characters as possible. As long as only characters 32-127 are used in the data, there is NO DIFFERENCE between data encoded in any ASCII format (including both ISO-8859-1 and Windows-

1252) and data encoded in UTF-8: each character is represented by one byte. But other UTF-8 characters are represented by two, three and four-byte strings. Some of these characters will have an equivalent in Windows-1252, and importers could include a translation routine for those characters. For other, non-translatable, characters, we recommend that importers translate these to blank, and issue a warning message to users that some data has been lost.

Note that the <position> element for fixed format data specifies the “character” position in the data, not the “byte” position. This distinction was not significant with the 8-bit single byte data characters used in ASCII encodings such as ISO-8859-1 and Windows-1252. But if the data encoding is multi-byte (as in UTF-8) then this is no longer true, so exporters and importers must be careful to use the “character” position.

Data encoded in UTF-8 may, optionally, include a three-byte Byte Order Mark (BOM) at the beginning of the file. Exporters may include this mark in their files if they wish. Importers should note that these bytes may be present, and should ignore them if they are (i.e. not see them as part of the data).

## Rich Text

Some implementers have for some time been exporting “rich” (i.e. formatted) text in Triple-S metadata files, using various methods, and this has occasionally caused problems for importers. From version 3.0, there is a Triple-S method for exporting rich text, which importers should recognise and can act upon as they wish, including ignoring such formatting commands completely. If an exporter’s software supports text formatting, and they want that formatting to be available to an importer, then if the formatting commands are not in (any version of) html/xhtml they should first translate them into html/xhtml . They should then incorporate the text and formatting string(s) within the CDATA markers “<![CDATA[“ and ”]]>”. So if a text string is, for example

“Less than 21 years old”

then an exporter might (n.b. there are various ways of representing underline in html) show this as

```
<![CDATA[Less than 21 <u>years</u> old]]>
```

An importer should begin by removing the CDATA markers from the text. Within what remains, they should then translate any format marking that their system can handle, and ignore the tags of those elements that it cannot.

The formatted html/xhtml texts can include references to inline, or internal, or external Cascading Style Sheets. Inline CSS may be introduced as style attributes of suitable text encapsulation elements in the body of the formatted text.

```
<![CDATA[Less than 21 <span style="text-decoration: underline">years</span> old]]>
```

As an alternative, references can be made via the “class” attribute to style components defined in internal or external CSS style sheets.

```
<![CDATA[Less than 21 <span class="underline">years</span> old]]>
```

These CSS style sheets are specified by one or more <style> elements, placed immediately before the <survey> element. For an internal style sheet the CSS code is defined within the body of the <style> element, whilst an external style sheet is specified by an “href” attribute.

```
<style href="myExtraStyles.css"/>
<style>
  .underline {
    text-decoration: underline;
  }
</style>
```

## Multiple keys for hierarchies

Until now, it has only been possible to link Triple-S metadata files which form a hierarchy by means of one variable. At Triple-S version 3.0, more than one (quantity or text) variable may be specified.

The change is reflected in the specification of the “linkvar” element in the hierarchical metadata file: this may now consist of two or more variable names, separated by commas. Each variable in the list must appear in both parent and child metadata file.

*For a parent and a child to match, each of the link variables in the child must compare equal to the corresponding variable in the parent. Note that whether or not they compare equal will be affected by the type of variable (for instance "0091" matches " 91" in a **quantity** but not a **character** variable.*

Every record in the child data set must have a parent in the higher level set. Orphan lower level data is illegal. It is, however, not necessary for each parent to have any children.

## Identifying the default language

Although it has been possible since Triple-S version 1.2 to specify that texts in different languages may appear in various Triple-S elements, by using the “languages” attribute in the initial <sss> element, there has been no way of indicating which the default language is.

To fill this omission, at Triple-S Version 3.0 it is possible to include an “xml:lang” attribute in the opening <sss> element in the file, for example

```
<sss version="3.0" xml:lang="en-GB" languages="en-GB en-US fr">
```

This attribute may be present regardless of whether multiple languages are in fact used.

In the above example, where there are multiple languages, “en-GB” may be omitted from the list of “languages”.

The introduction of this feature reduces the number of “xml:lang” attributes that are needed in text elements in the case where there are multiple languages present. In the following sequence

```
<value code="1">Yes
  <text xml:lang="en-GB">Yes</text>
  <text xml:lang="en-US">Sure</text>
  <text xml:lang="fr">Oui</text>
</value>
```

The first text element, necessary until version 3.0, may now be omitted.

## Scores on multi-response variables

Up until now Triple-S has only allowed “score” attributes, used to indicate the value to be used when computing statistics such as Mean, Standard Deviation etc. for the variable, on single variables.

Scored variables are, however, often defined at source as multiples, containing one or more “net” bits, typically “top” and “bottom” “boxes”, which are summaries for those respondents who checked (e.g.) either of the top two or bottom two points on a rating scale. Up until now, it has not been possible to export the scores for these variables.

At Triple-S version 3.0, “score” attributes are also allowed on variables of type multiple.

The omission of a score, on both single and multiple variables, implies that records having that value code *and no other value codes* might be omitted from the base for any statistical computation for that variable.

## Value code 0 and literal codes in spread multiples

Until now, value codes for multiple variables had to be positive integers. From Triple-S Version 3.0, if the <spread> element has been specified for a multiple variable, then the value code “0” is also allowed.

If the <spread> element has been specified for a multiple variable, then value codes may include literal values (as for single variables), providing the attribute <format=literal> is included in the <variable> element.

Note that if 0 is used as a value then its meaning will not be ‘not used’, as is the case with bitstring multis. And note that “ ” (blank) is not allowed as a literal code in any case.