

The data schema used by DdlUtils

Table of contents

1 The data schema used by DdlUtils.....	2
---	---

1. The data schema used by DdlUtils

DdlUtils uses a dynamic XML schema for representing data that tries to use table and column names as much as possible. It follows the following rules for choosing the tags and attributes to represent the table of a given row:

- If the table name is a valid XML tag, then it will be used as the XML element for the row. E.g.

```
<MyTable ...>...</MyTable>
```

- If the table name is not a valid XML tag, but it is shorter than 255 characters and does not contain characters that would be illegal for an XML attribute, then the XML element will be `table` with an attribute `table-name` containing the name of the table. E.g.

```
<table table-name="My Table" ...>...</table>
```

If the table name contains characters like the ampersand, then these will be escaped in the standard XML fashion (via entities):

```
<table table-name="Bread&Butter" ...>...</table>
```

- If the table name is not a valid XML tag (not a valid tag or longer than 255 characters) and does not contain characters that would be illegal for an XML attribute, then the XML element will be `table` with a sub element `table-name` containing the name of the table. E.g.

```
<table ...>
  <table-name>My Really Long Table Name ...</table-name>
  ...
</table>
```

- If the table name contains characters that are illegal in XML, then the same format as above is used, but the value is also Base-64 encoded. An additional attribute `base64` with the value `true` signifies that the value is Base-64 encoded. E.g.

```
<table ...>
  <table-name base64="true">TXlUYWJsZQ==</table-name>
  ...
</table>
```

The rules for the columns are similar:

- If the column name is a valid XML attribute name and not equal to `table-name` or `base64`, and the value is shorter than 255 characters and does not contain any characters invalid in XML, then an XML attribute will be used for the column. This is true regardless of whether the table name is a valid tag:

```
<MyTable myColumn="..." ...>...</MyTable>
```

or not:

```
<table table-name="My Table" myColumn="..." ...>...</table>
```

- If the column name is a valid XML attribute name and not equal to `table-name` or

base64, but the value is not shorter than 255 characters or it contains characters that are not allowed in XML documents, then instead a sub element will be used with the column name as the tag name:

```
<MyTable ...>
  <myColumn>...</myColumn>
  ...
</MyTable>
```

or

```
<MyTable ...>
  <myColumn base64="true">...</myColumn>
  ...
</MyTable>
```

if the value needs to be Base-64 encoded because of illegal characters.

- If the column name is not a valid XML attribute name and it is shorter than 255 characters and does not contain characters that would be illegal for an XML attribute, or if the column name is equal to `column-name` or `base64`, then instead a sub element will be used for the column name which will have an attribute `column-name` for the column name and the value as text content: E.g.

```
<MyTable ...>
  <column column-name="the column">...</column>
  ...
</MyTable>
```

or

```
<MyTable ...>
  <column column-name="the column" base64="true">...</column>
  ...
</MyTable>
```

if the value needs to be Base-64 encoded.

- If the column name is not a valid XML attribute name or it is longer than 255 characters or it contains illegal characters, then instead a `column-name` sub element is used with the column name as the text content (base64 encoded if necessary). The value will be in a corresponding `column-value` sub element:

```
<MyTable ...>
  <column>
    <column-name>...</column-name>
    <column-value>...</column-value>
  </column>
  ...
</MyTable>
```