

8.21 Storage Size

8.21.1 Definition



The **Storage Size** data category is used to specify the maximum storage size of a given content.

This data category can be used for various purposes, including the following examples:

- Verify during translation if a string fits into a fixed-size database field.
- Control the size of a string that is stored in a fixed-size memory buffer at run-time.

The storage size is always expressed in bytes and excludes any Byte-Order-Marks. It is and is provided along with the character set encoding and the line break which will be used when the content is stored to store the content. If the encoding form does not use the byte as its unit (e.g. UTF-16 uses 16-bit code units) the storage size MUST still be given in byte (e.g. for UTF-16: 2 bytes per 16-bit code unit).

An application verifying the storage size for a given content is expected to perform the following steps:

- All the LINE FEED (U+000A) characters of the content to verify are replaced by the character or characters specified by the line break type.
- The resulting string is converted to an array of bytes using a character set encoder for the specified encoding.
- If the leading bytes represent a Byte-Order-Mark, they are stripped from that array.
- The length of the resulting array is compared to the storage size provided. The content is too long if the length is greater than the storage size.

Note: Storage size is not related to the display length of a text, and therefore should not be used to constrain a certain display length.

8.21.2 Implementation



The **Storage Size** data category can be expressed with global rules, or locally on individual elements. There is no inheritance. The default value of the

Formatted: Font: (Default) Arial, (Asian) Times New Roman, 13.5 pt, Font color: Black

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Font: (Default) Arial, (Asian) Times New Roman, 13.5 pt, Font color: Black

Formatted: Font: (Default) Arial, (Asian) Times New Roman, 13.5 pt, Font color: Black

character `set` encoding is UTF-8 and the default value for the line break is lf (LINE FEED (U+000A)).

GLOBAL: The `storageSizeRule` element contains the following:

- A required `selector` attribute. It contains an absolute selector which selects the nodes to which this rule applies.
- Exactly one of the following:
 - A `storageSize` attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.
 - A `storageSizePointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `storageSize`.
- None or exactly one of the following:
 - A `storageEncoding` attribute. It contains the name of the character `set` encoding used to calculate the number of bytes of the selected text. The name **MUST** be one of the names or aliases listed in the IANA Character Sets registry [IANA Character Sets]. The default value is the string "UTF-8".
 - A `storageEncodingPointer` attribute that contains a relative selector pointing to a node with the exact same semantics as `storageEncoding`.
- An optional `lineBreakType` attribute. It indicates what type of line breaks the storage uses. The possible values are: `cr` for CARRIAGE RETURN (U+000D), `lf` for LINE FEED (U+000A), or `crlf` for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A), ~~or `nel` for NEXT LINE (U+0085)~~. The default value is `lf`.

Example 92: The Storage Size data category expressed globally in XML

The `storageSizeRule` element is used to specify that, when encoded in ISO-8859-1, the content of the `country` element must not be more than 25 bytes. The name "Papouasie-Nouvelle-Guinée" is 25 character long and fits because all characters in ISO-8859-1 are encoded as a single byte.

```
<?xml version="1.0"?>
<db>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:storageSizeRule selector="//country" storageSize="25">
```

```

        storageEncoding="ISO-8859-1"/>

</its:rules>

<data>

  <country id="123">Papouasie-Nouvelle-Guinée</country>

  <country id="139">République Dominicaine</country>

</data>

</db>

```

[Source file: [examples/xml/EX-storageSize-global-1.xml](#)]

Example 93: Mapping the [Storage Size](#) data category in XML

The `storageSizePointer` attribute is used to map the non-ITS attribute `max` to the same functionality as `storageSize`. There is no character encoding specified, so the default UTF-8 is assumed. Note that, while the name "Papouasie-Nouvelle-Guinée" is 25 character long, the character 'é' is encoded into two bytes in UTF-8. Therefore this name is one byte too long to fit in its storage destination.

```

<?xml version="1.0"?>

<fields>

  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">

    <its:storageSizeRule selector="//field" storageSizePointer="@max"/>

  </its:rules>

  <field type="country" id="123" max="25">Papouasie-Nouvelle-
Guinée</field>

  <field type="country" id="139" max="25">République Dominicaine</field>

</fields>

```

[Source file: [examples/xml/EX-storageSize-global-2.xml](#)]

LOCAL: the following local markup is available for the [Storage Size](#) data category:

- A `storageSize` attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.

- An optional `storageEncoding` attribute. It contains the name of the character encoding used to calculate the number of bytes of the selected text. The name **MUST** be one of the names or aliases listed in the [IANA Character Sets registry \[IANA Character Sets\]](#). The default value is "UTF-8".
- An optional `lineBreakType` attribute. It indicates what type of line breaks the storage uses. The possible values are: `cr` for CARRIAGE RETURN (U+000D), `lf` for LINE FEED (U+000A), or `crlf` for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A), ~~or `nel` for NEXT LINE (U+0085)~~. The default value is `lf`.

Example 94: The [Storage Size](#) data category expressed locally in XML

The `storageSize` attribute allows to specify different ~~the~~ maximum storage sizes throughout the document. Note that the string CONTINUE does not fit the specified restriction of 8 bytes. The minimal number of bytes to store such a string in UTF-16 is 16.

```
<?xml version="1.0"?>
<messages xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <var num="panelA1_Continue" its:storageSize="8"
its:storageEncoding="UTF-16">CONTINUE</var>
  <var num="panelA1_Stop" its:storageSize="8" its:storageEncoding="UTF-
16">STOP</var>
  <var num="panelB5_Cancel" its:storageSize="12"
its:storageEncoding="UTF-16">CANCEL</var>
</messages>
```

[Source file: [examples/xml/EX-storageSize-local-1.xml](#)]

Note:

~~Please note that the string CONTINUE in the above example does not fit the specified restriction.~~

Example 95: The [Storage Size](#) data category expressed locally in HTML

The `its-storage-size` is used here to specify the maximum number of bytes the two editable strings can have in UTF-8.

```
<!DOCTYPE html>
<html lang=en>
```

```
<head>
  <meta charset=utf-8>
  <title>Example</title>
</head>
<body>
  <p>String to translate:</p>
  <p contenteditable=true id=123 its-storage-size=25>Papua New-
Guinea</p>
  <p contenteditable=true id=139 its-storage-size=25>Dominican
Republic</p>
</body>
</html>
```

[Source file: <examples/html5/EX-storageSize-html5-local-1.html>]