

MODEL Text

The present model defines a set of standard functions for the data types TEXT and MTEXT, as a basis for further consistency conditions.

compareToIgnoreCase()

Compares the two arguments lexically, ignoring upper and lower case.

```
FUNCTION compareToIgnoreCase(a: TEXT; b: TEXT): NUMERIC;  
FUNCTION compareToIgnoreCaseM(a: MTEXT; b: MTEXT): NUMERIC;
```

concat()

Joins the two arguments together.

```
FUNCTION concat(a: TEXT; b: TEXT): TEXT;  
FUNCTION concatM(a: MTEXT; b: MTEXT): MTEXT;
```

endsWith()

Checks whether the end of the first argument matches the second argument.

```
FUNCTION endsWith(val: TEXT; suffix: TEXT): BOOLEAN;  
FUNCTION endsWithM(val: MTEXT; suffix: MTEXT): BOOLEAN;
```

equalsIgnoreCase()

Checks whether the two arguments are lexically equal, ignoring upper/lower case.

```
FUNCTION equalsIgnoreCase(val: TEXT; anotherVal: TEXT): BOOLEAN;  
FUNCTION equalsIgnoreCaseM(val: MTEXT; anotherVal: MTEXT):  
BOOLEAN;
```

indexOf()

Returns the index where the second argument occurs in the first argument or -1 if it does not occur. If fromIndex is defined, the search starts from the given position.

```
FUNCTION indexOf(val: TEXT; str: TEXT; fromIndex: NUMERIC):  
NUMERIC;  
FUNCTION indexOfM(val: MTEXT; str: MTEXT; fromIndex: NUMERIC):  
NUMERIC;
```

lastIndexOf()

Returns the index where the second argument occurs in the first argument or -1 if it does not occur. The search begins at the end. If fromIndex is defined, the search starts from the given position.

```
FUNCTION lastIndexOf(val: TEXT; str: TEXT; fromIndex: NUMERIC):  
NUMERIC;  
FUNCTION lastIndexOfM(val: MTEXT; str: MTEXT; fromIndex:  
NUMERIC): NUMERIC;
```

matches()

Checks whether the first argument matches the regular expression.

```
FUNCTION matches(val: TEXT; regex: TEXT): BOOLEAN;
FUNCTION matchesM(val: MTEXT; regex: TEXT): BOOLEAN;
```

replace()

Replaces with the third argument the places where the second argument occurs in the first argument.

```
FUNCTION replace(val: TEXT; old: TEXT; new: TEXT): TEXT;
FUNCTION replaceM(val: MTEXT; old: MTEXT; new: MTEXT): MTEXT;
```

startsWith()

Checks whether beginning of the first argument matches the second argument.

```
FUNCTION startsWith(val: TEXT; prefix: TEXT): BOOLEAN;
FUNCTION startsWithM(val: MTEXT; prefix: MTEXT): BOOLEAN;
```

substring()

Returns from the first argument the range defined by the second and third arguments.

```
FUNCTION substring(val: TEXT; beginIndex: NUMERIC; endIndex:
NUMERIC): TEXT;
FUNCTION substringM(val: MTEXT; beginIndex: NUMERIC; endIndex:
NUMERIC): MTEXT;
```

toLowerCase()

Returns the text in lower case.

```
FUNCTION toLowerCase(val: TEXT): TEXT;
FUNCTION toLowerCaseM(val: MTEXT): MTEXT;
```

toUpperCase()

Returns the text in upper case.

```
FUNCTION toUpperCase(val: TEXT): TEXT;
FUNCTION toUpperCaseM(val: MTEXT): MTEXT;
```

Attachment A

INTERLIS 2.3;

CONTRACTED TYPE MODEL Text (en) AT "http://www.interlis.ch/models"
VERSION "19/11/2018" =

```
FUNCTION compareToIgnoreCase(a: TEXT; b: TEXT): NUMERIC;
FUNCTION compareToIgnoreCaseM(a: MTEXT; b: MTEXT): NUMERIC;
FUNCTION concat(a: TEXT; b: TEXT): TEXT;
FUNCTION concatM(a: MTEXT; b: MTEXT): MTEXT;
```

```
FUNCTION endsWith(val: TEXT; suffix: TEXT): BOOLEAN;
FUNCTION endsWithM(val: MTEXT; suffix: MTEXT): BOOLEAN;
FUNCTION equalsIgnoreCase(val: TEXT; anotherVal: TEXT): BOOLEAN;
FUNCTION equalsIgnoreCaseM(val: MTEXT; anotherVal: MTEXT):
BOOLEAN;
```

```
    FUNCTION indexOf(val: TEXT; str: TEXT; fromIndex: NUMERIC):
NUMERIC;
    FUNCTION indexOfM(val: MTEXT; str: MTEXT; fromIndex: NUMERIC):
NUMERIC;
    FUNCTION lastIndexOf(val: TEXT; str: TEXT; fromIndex: NUMERIC):
NUMERIC;
    FUNCTION lastIndexOfM(val: MTEXT; str: MTEXT; fromIndex:
NUMERIC): NUMERIC;
    FUNCTION matches(val: TEXT; regex: TEXT): BOOLEAN;
    FUNCTION matchesM(val: MTEXT; regex: TEXT): BOOLEAN;
    FUNCTION replace(val: TEXT; old: TEXT; new: TEXT): TEXT;
    FUNCTION replaceM(val: MTEXT; old: MTEXT; new: MTEXT): MTEXT;
    FUNCTION startsWith(val: TEXT; prefix: TEXT): BOOLEAN;
    FUNCTION startsWithM(val: MTEXT; prefix: MTEXT): BOOLEAN;
    FUNCTION substring(val: TEXT; beginIndex: NUMERIC; endIndex:
NUMERIC): TEXT;
    FUNCTION substringM(val: MTEXT; beginIndex: NUMERIC; endIndex:
NUMERIC): MTEXT;
    FUNCTION toLowerCase(val: TEXT): TEXT;
    FUNCTION toLowerCaseM(val: MTEXT): MTEXT;
    FUNCTION toUpperCase(val: TEXT): TEXT;
    FUNCTION toUpperCaseM(val: MTEXT): MTEXT;

END Text.
```