

MODEL Text

Mit dem vorliegenden Modell wird eine Menge von Standard-Funktionen im Bereich der Datentypen TEXT und MTEXT definiert, als Basis für weitergehende Konsistenzbedingungen.

compareToIgnoreCase()

Vergleicht die beiden Argumente lexikalisch, und ignoriert dabei Gross/Kleinschreibung.

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

concat()

Fügt die beiden Argumente zusammen/aneinander.

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

endsWith()

Prüft, ob das Ende des ersten Argumentes, dem zweiten Argument entspricht.

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

equalsIgnoreCase()

Prüft, ob die beiden Argumente lexikalisch gleich sind, und ignoriert dabei Gross/Kleinschreibung.

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

indexOf()

Liefert den Index, an dem das zweite Argument im ersten Argument vorkommt oder -1, falls es nicht vorkommt. Wenn fromIndex definiert ist, beginnt die Suche ab der gegebenen Position.

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

lastIndexOf()

Liefert den Index, an dem das zweite Argument im ersten Argument vorkommt oder -1, falls es nicht vorkommt. Die Suche beginnt von hinten. Wenn fromIndex definiert ist, beginnt die Suche ab der gegebenen Position.

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

matches()

Prüft, ob das erste Argument dem regulären Ausdruck entspricht.

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

replace()

Ersetzt im ersten Argument die Stellen bei denen das zweite Argument vorkommt, mit dem dritten Argument.

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

startsWith()

Prüft, ob der Anfang des ersten Argumentes, dem zweiten Argument entspricht.

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

substring()

Liefert aus dem ersten Argument den Bereich, der durch das zweite und dritte Argument definiert wird.

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

toLowerCase()

Liefert den Text in Kleinbuchstaben.

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

toUpperCase()

Liefert den Text in Grossbuchstaben.

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

Anhang A

INTERLIS 2.3;

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

```
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.
```