|  |
| --- |
|  |

***Konfiguration***

***Version 1.0***

***18. Juli 2019***

**Konventionen**

In dieser Dokumentation werden die folgenden Konventionen verwendet um den Überblick und das schnelle Erfassen der gelieferten Informationen zu erleichtern.

**Beispiel**



Angabe eines Beispiels, das die zuvor beschriebene Eigenschaft bzw. Funktion verdeutlicht.

**Tipp**



Angabe eines Tipps, wie eine beschriebene Funktionalität eingesetzt werden kann.

**Arbeitsweise**



Beschreibung der internen Arbeitsweise so dass ein besseres Verständnis erreicht wird.

**Verbot**



Angabe von verbotenen oder nicht unterstützten Eigenschaften oder Funktionen.

**Frage**



Fragen und Antworten die in dem betreffenden Zusammenhang bereits gestellt wurden.

**Links**



Hinweise auf Internetseiten die weitere Informationen beinhalten.

**Hinweis**



Hinweis, der beim Einsatz beachtet werden muss.

**Achtung**



Wichtiger Hinweis der zum fehlerfreien Einsatz der Funktionalität unbedingt beachtet werden muss.

**Inhalt**

[Kapitel 3](#_Toc132508842)

# Allgemeines

Die Konfigurationskomponente soll für alle zukünftigen Anwendungen zur Verfügung stehen. Sie kapselt alle Details des tatsächlichen Speicherorts der Konfigurationsdaten.

Die Definition welche Elemente konfiguriert werden können sind in Typbibliotheken gespeichert (XML-Format) die beim Start eingelesen werden.

## Designziele

* Sprachumschaltung des User-Interfaces
* Bei Änderungen sollen zu Vergleichszwecken die vorherigen und die Standardwerte angezeigt werden
* Werte sollen schon bei der Eingabe auf Gültigkeit geprüft werden

# Softwarestruktur

# Konfigurationsbaum

Die Konfigurationsdaten sind hierarchisch strukturiert. Die Struktur ergibt sich aus den Definitionen in der Typ-Bibliothek und den Einträgen der Datenbank.

# Benutzer und Sicherheit

## Benutzerverwaltung

Die Benutzerverwaltung und Vergabe der Rechte erfolgt nach folgenden Punkten.

* Rechte können nur für Claims (Gruppen bzw. Rollen) vergeben werden
* Benutzern können mehrere Claims zugeordnet werden

## Die Vergabe von Rechten

Rechte sind pro Claim getrennt Administrierbar. Untergeordnete Zweige erben automatisch alle Rechte der übergeordneten Ebene wenn keine anderen Rechte vergeben wurden.

* Vererbt
* Anzeigen
* Ändern
* Hinzufügen
* Löschen

# Definitionen

## Konfigurationseinheiten

### ID

Eindeutige ID des Element-Typs.

### DataType

Code des Datentyps den dieses Element speichert.

#### DataType.Object

Dieser Datentyp gibt an, dass es sich um einen Knoten handelt, der im Konfigurationsbaum angezeigt wird. Er kann wahlweise Eigenschaften und/oder Unterknoten enthalten.

#### DataType.Empty

Ein Element dieses Typs dient als Platzhalter für benutzerdefinierte Elemente. Sie spezifizieren die Struktur dieser Elemente.

### DisplayName

Text zur Anzeige des Elements an der Programmoberfläche. Dieser Text kann landessprachig angepasst werden.

### DisplayImage

Index des Bildes, das für dieses Element angezeigt werden soll.

Die Bilder sind in einer ImageList des Konfigurations-Explorers abgelegt, stehen jedoch auch im Ordner Images als GIF Bilder (*Index*.GIF) zur Verfügung.

(siehe Symbole)

### Cathegory

Text der als Gruppierung von Eigenschaftswerten in der Property-Liste verwendet wird.

### Description

Kurze Beschreibung des Elements. Diese wird bei der Auswahl des Elements an der entsprechenden Oberfläche angezeigt werden.

### HelpUrl

Link zur ausführlichen Hilfe zu diesem Element.

### DefaultValue

Standardwert der verwendet wird, wenn kein Eintrag in der Konfigurationsdatenbank vorhanden ist.

Die Konfigurationsoberfläche sollte eine Möglichkeit bieten einen Wert auf diesen Standardwert zurück zu setzen.

### Unit

Einheit des Wertes (z.B. Sekunden).

Endet die Einheit mit einem Leerzeichen so wird die Einheit nur in der Hilfe angezeigt.

### InputType

Gibt an ob der Parameter optional oder erforderlich ist. Dies wird in Konfigurationsoberfläche farblich gekennzeichnet.

0 – (Standard) optionale Angabe

1 – zwingende Angabe (für die Funktion der Software erforderlich)

### SelectionType

Auswahl der Art wie der Eigenschaftswert bearbeitet werden kann.

### ValidationRule

VB.NET Ausdruck der die Gültigkeit des Wertes prüft.

**Hinweis**



Der momentane Eingabewert steht als **value** zur Verfügung.

**Beispiel**



Prüfung auf Wertebereich 0..99:

(value >= 0) AND (value <= 99)

### ValidationMessage

Fehlertext der angezeigt wird, wenn der Eingabewert die Gültigkeitsregel verletzt.

### NamespaceProvider

Angabe wie die Einstellungen gespeichert werden.

file://<FileName>

db://<ConnectionString>

schema://<SchemaName>

memory://

### ValueList

Liste von mit Strichpunkt getrennten Wertepaaren die zur Auswahl verwendet werden.

Das Wertepaar besteht immer aus Wert und Anzeigetext.

z.B.: 0;Falsch;1;Wahr

oder: 1;eine Sekunde;10;zehn Sekunden;60;eine Minute

**Auswahl mit Einheit**

Endet die Auswahl mit einem Leerzeichen so wird die Einheit nicht angezeigt.

<Unit>Minuten</Unit>

<SelectionType>7</SelectionType>

<ValueList>.5;.5 (30 Sekunden) ;1;1 Minute ;2;2;5;5</ValueList>

Die Einheit wird hier nur bei der Auswahl 2 und 5 oder der Eingabe eines nicht in der Liste enthaltenen Wertes (.75) angezeigt.

.5 (30 Sekunden)

.75 **Minuten**

1 Minute

2 **Minuten**

5 **Minuten**

### UserLevel

Dient zur Unterdrückung von speziellen Einstellungen die nur von erfahrenen Anwendern benötigt werden.

0 – Alle Anwender

1..n – erfahrenere Anwender

## Tabellenstruktur Konfigurationsdatenbank

## Aufzählungen

### Selektionstypen

|  |  |
| --- | --- |
| **Bezeichner** | **Beschreibung** |
| Parent | Element dient als Eltern für benutzerdefinierte Elemente. |
| Edit | Der Wert wird eingegeben. |
| EditPassword |  |
| EditScript | Script Editor |
| Boolean | Es wird eine Checkbox angezeigt. |
| ListOnly | Es wird aus einer Liste vorgegebener Werte ausgewählt (ValueList). |
| ListEdit | Der Wert wird eingegeben oder aus einer Liste vorgegebener Werte ausgewählt (ValueList). |
| SelectBits | Bitweise Optionen (ValueList enhält Definitionen). |
| Color | Farbauswahl (RGB-Wert wird in Datentyp UI4 abgelegt) |
| Reference | Es wird eine Referenz auf ein bestehendes Element ausgewählt. |
| ReferenceList | Es wird eine Liste von Referenzen auf ein bestehendes Element ausgewählt. |
| DbConnectString | ConnectString für Datenbankverbindungen |
|  |  |
| SELTYPE\_Path |  |
| SELTYPE\_ExistingPath |  |
| SELTYPE\_FileName |  |
| SELTYPE\_ExistingFileName |  |
| SELTYPE\_FileWildcard |  |
| SELTYPE\_PathWildcard |  |
| SELTYPE\_Date | Datumsangabe |
| SELTYPE\_ReferenceEdit | Es wird eine Liste von Referenzen auf ein bestehendes Element angeboten. ValueList enthält die ID des auszuwählenden Typs.  Wahlweise kann ein Element ausgewählt oder dessen Bezeichnung eingegeben werden. Gespeichert wird die Bezeichnung nicht die Referenz!!! Die Auflösung erfolgt erst zur Laufzeit. |

### Reference

ValueList enthält entweder eine Liste der IDs von möglichen VT\_OBJECT Elementen oder die ID eines VT\_FOLDER Elements dessen ValueList verwendet werden soll.

Wenn der Wert nicht zwingend ist wird die Auswahl um den Punkt „<Standard>“ ergänzt.

### EditList

ValueList enthält eine Liste mit möglichen Auswahlen. Wird auch Unit angegeben, so wird dieser hinter der Auswahl angezeigt. Um dies zu verhindern muss die Auswahl mit einem Leerzeichen enden.

### SelectBits

ValueList enthält eine Liste mit möglichen Auswahlen. Hierbei wird jede Auswahl als Option zum ankreuzen angeboten. Auf diese Weise können mehrere Optionen angegeben werden.

Die ValueList enthält zu jeder möglichen Bitwertigkeit (1,2,4,8,16,32,...) einen entsprechenden Text. Wird eine Auswahl mit dem Wert Null angegeben, so wird dieser Text angezeigt wenn keine Option ausgewählt ist.

<ValueList>0;keine;1;Eingänge;2;Ausgänge</ValueList>

### DbListOnly, DbEditList

ValueList enthält die Spezifikation, aus welcher Datenquelle die Auswahlliste erzeugt wird.

DbCnnName;DbSource;DbFields;DbWhere;DbOrder;ValFmt

**DbCnnName**

Bezeichnung der Datenbankverbindung die verwendet werden soll (ohne $DB\_*Name*).

Es kann ein Name aus der Hierarchie der aktuellen Eigenschaft verwendet werden.

%PARENTNAME1% ... %PARENTNAME5% geben jeweils die Bezeichnung der übergeordneten Hierarchieebene an (PARENTNAME1 = eine Ebene höher, PARENTNAME2 = zwei Ebenen höher, usw.).

**DbSource**

Angabe der Tabelle oder Abfrage.

**DbFields**

Felder der Datenquelle die zur Bildung von Wert und Anzeigetext herangezogen werden.  
Ist der Anzeigetext gleich dem Wert muss genau ein Feld, ansonsten müssen genau zwei Felder (Wert und Anzeigetext) angegeben werden.

**DbWhere**

Bedingung, der die gelieferten Datensätze entsprechen müssen.

**DbOrder**

Angabe des Feldes nach dem die Auswahl sortiert warden soll.

**ValFmt**

Formatanweisung die angibt wie aus dem Wert aus der Datenbank der Wert für die Auswahl gebildet wird. Der Wert aus der Datenbank wird hierbei mit %VAL% angegeben.

**Beispiel**

<ValueList>%PARENTNAME4%;Masch;ID,Text;;Text;Masch=%VAL%</ValueList>

DbCnnName = Name des vierthöheren Elements

DbSource = “Masch“ (Bezeichnung der DB-Tabelle)

DbFields = “ID,Text“

DbWhere = leer (alle Datensätze)

DbOrder = “Text“ (aufsteigend sortiert nach Text)

ValFmt = “Masch=%VAL%“ (aus DB-Eintrag 5 wir der Wert “Masch=5“ der Auswahl)